20TETO*h6099001

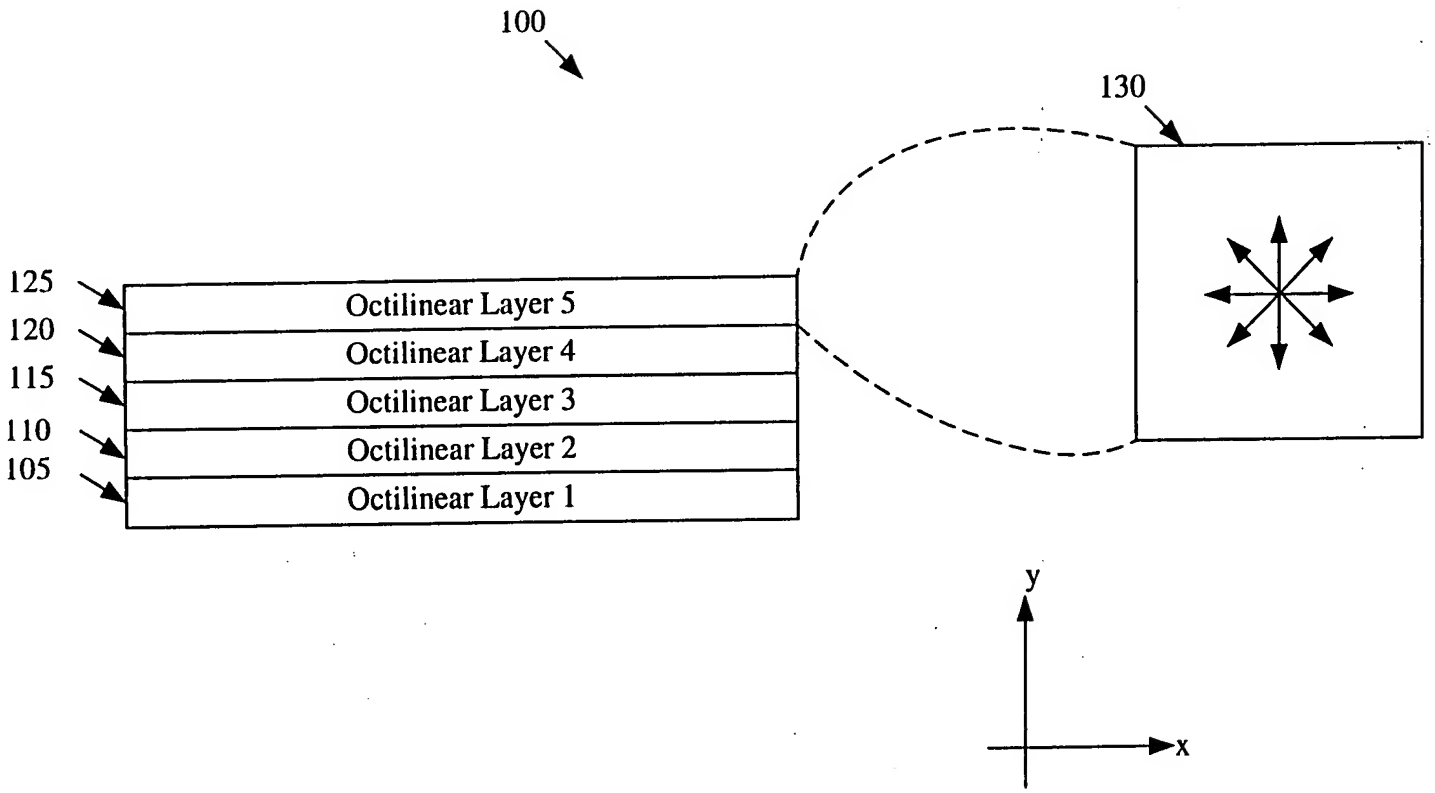


Figure 1

Docket No. 2003P04950US

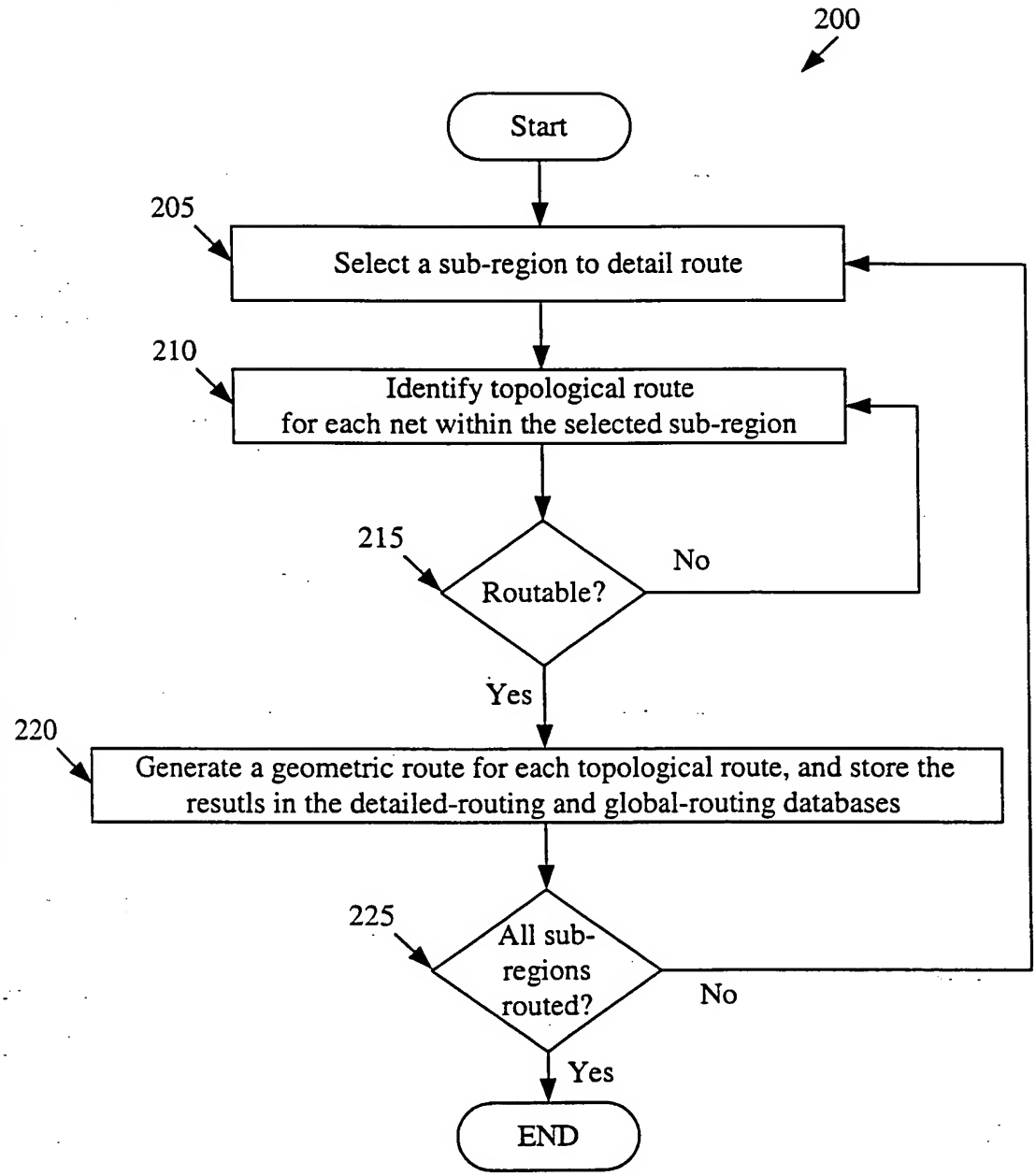


Figure 2

20120146099001

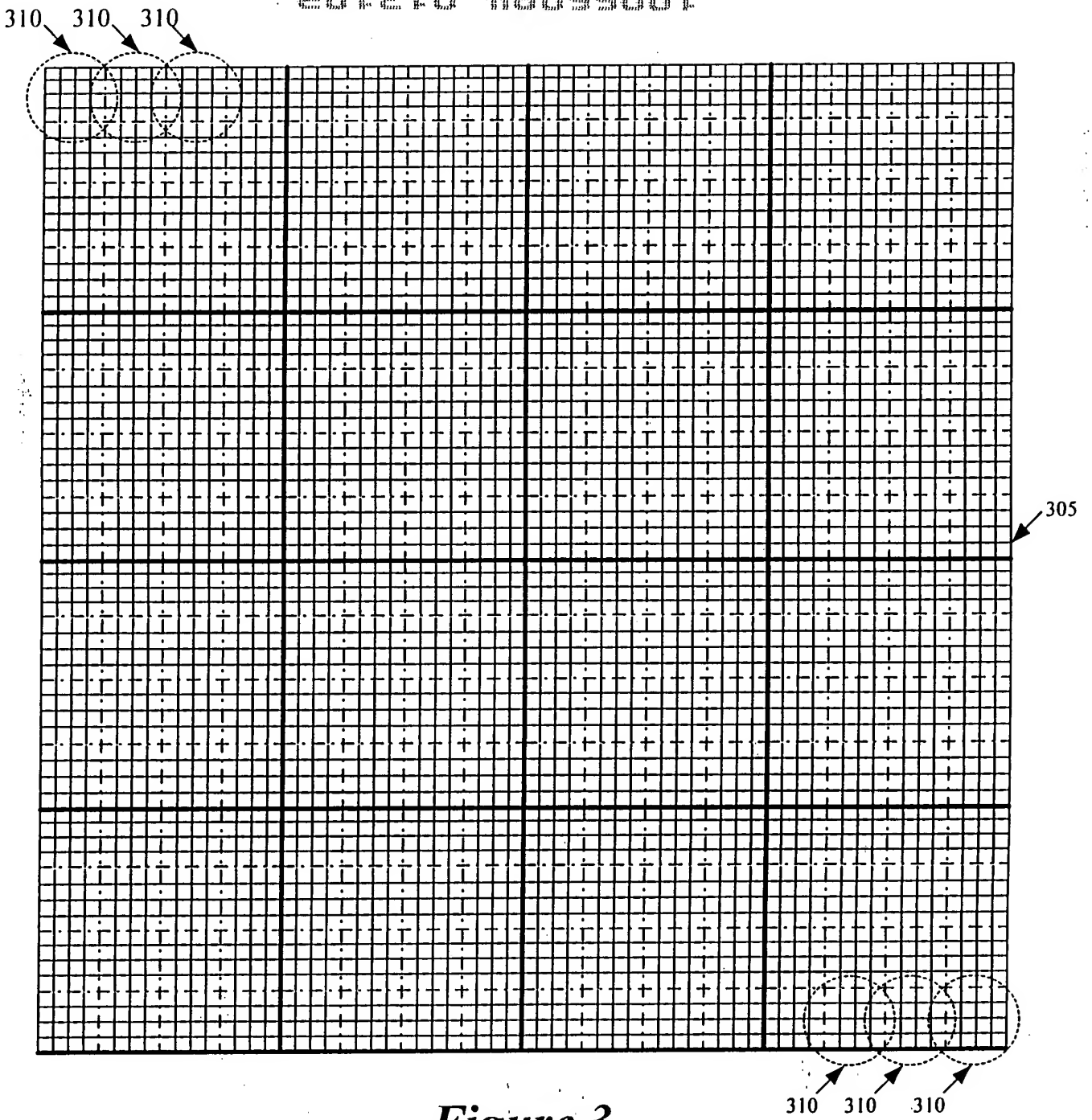


Figure 3

20020 4609901

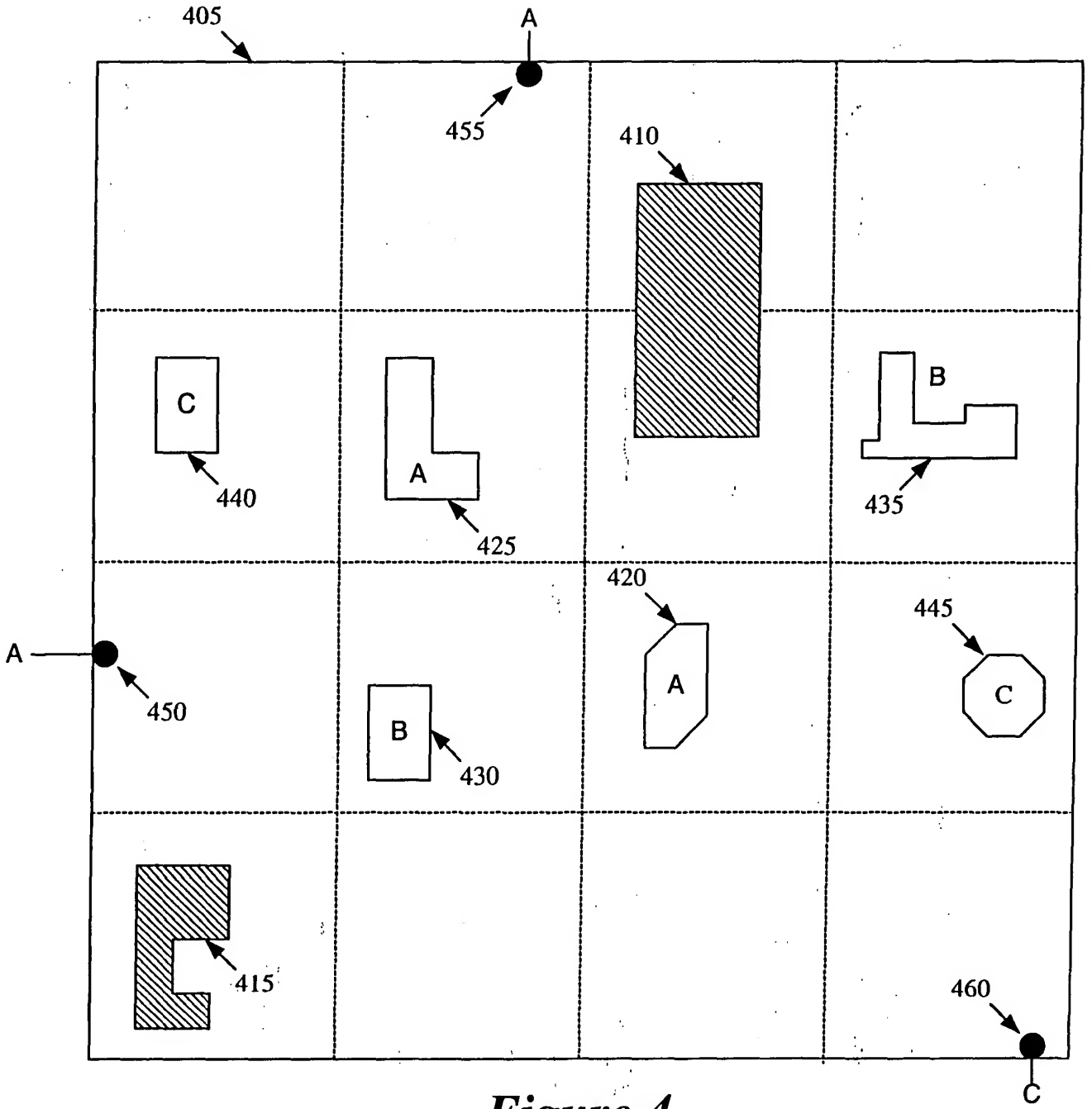


Figure 4

20TETO"h6099001

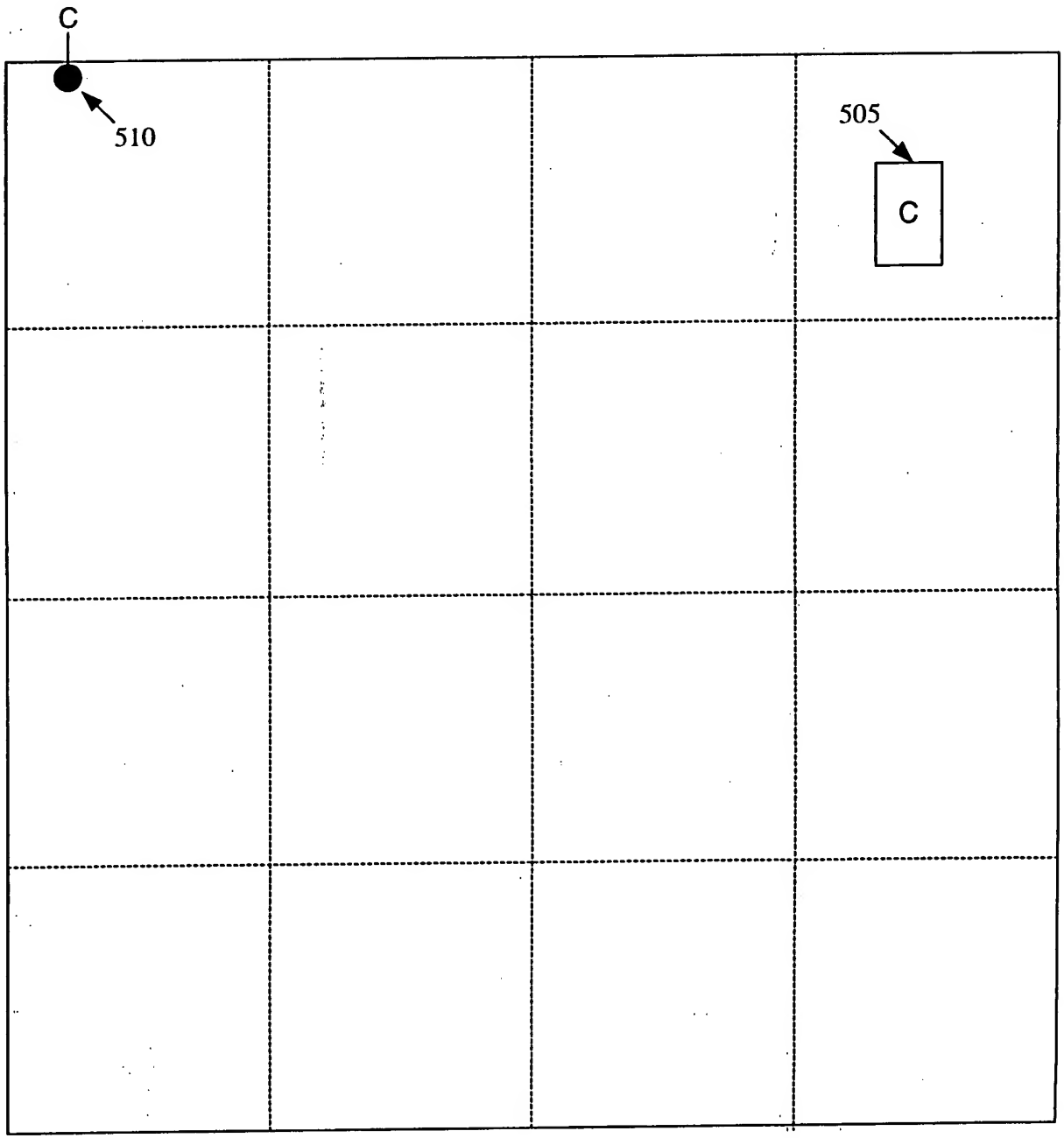


Figure 5

-List of Geometries
    --Each Geometry including a sequence of points & layer assignment
-Bounding box of the region
-Array of layer properties
    --Minimum wire size
    --Minimum spacing
    --Via sizes
    --Cost/Unit
-Netlist specifying a number of nets
    --Each net specifying a set of pins
        --Each pin specifying a set of ports
            --Each port specifying a set of geometries

Figure 6

-List of Geometries
    --Each Geometry including a sequence of points & layer assignment
        --List of connection nodes inside each pin geometry
-Bounding box of the region
-Array of layer properties
    --Minimum wire size
    --Minimum spacing
    --Via sizes
    --Cost/Unit
-Netlist specifying a number of nets
    --Each net specifying a set of pins
        --Each pin specifying a set of ports
            --Each port specifying a set of geometries
-For each layer, a graph specifying
    --Nodes
    --Edges
    --Faces

Figure 7

20TETO"H6099001

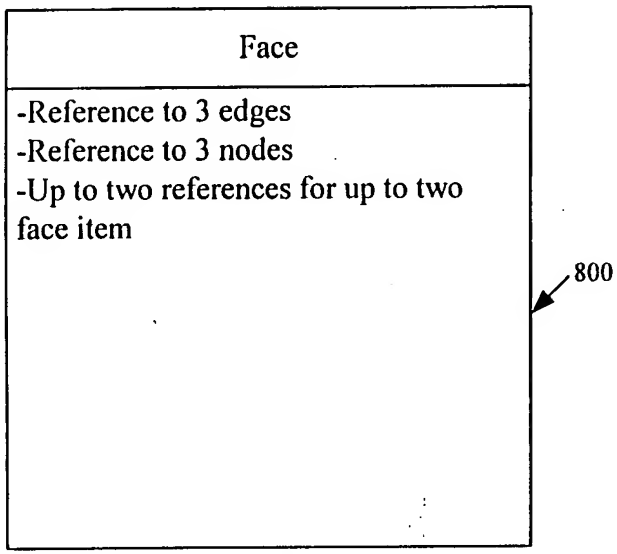


Figure 8

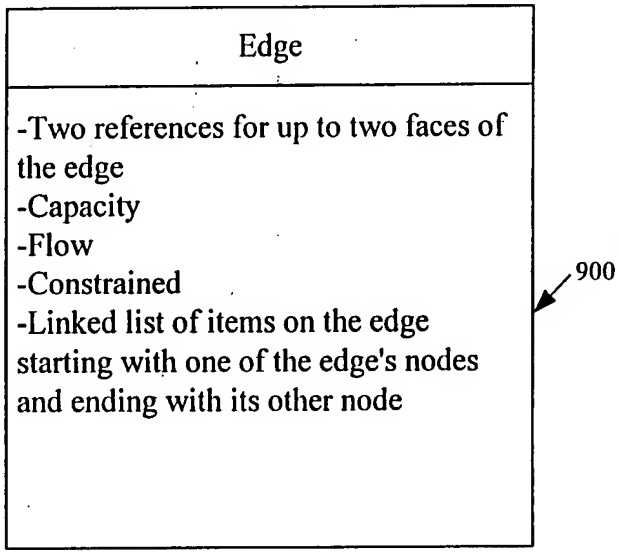


Figure 9

20020704.160939OT

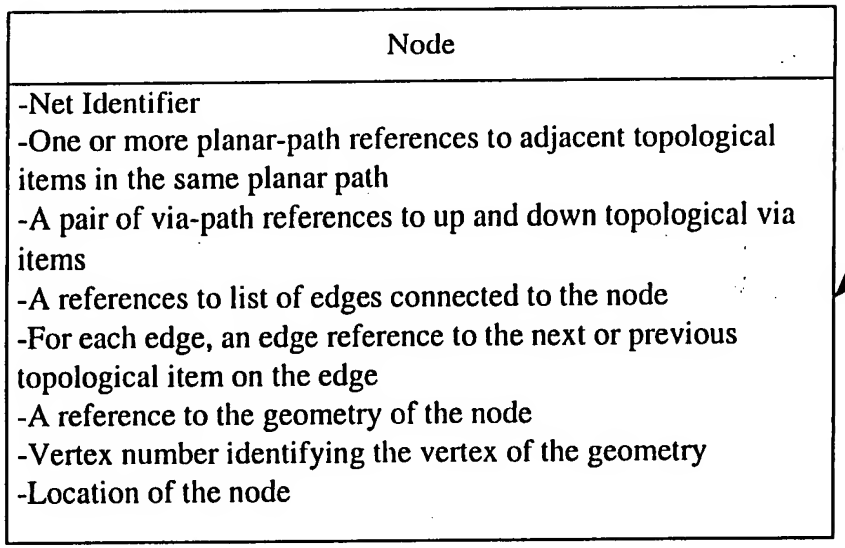


Figure 10

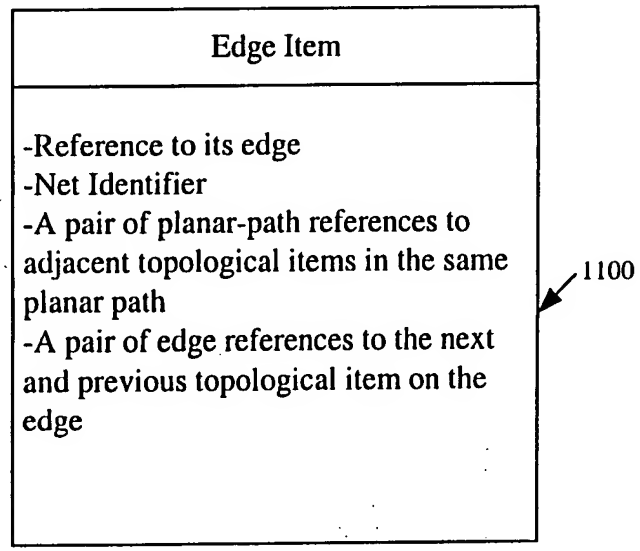


Figure 11

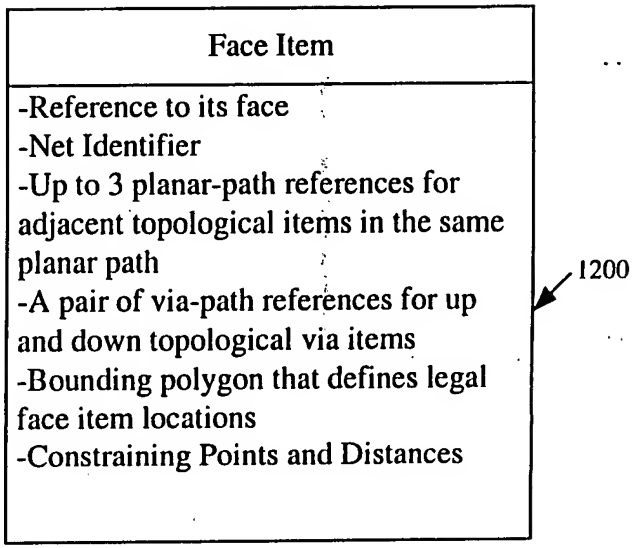
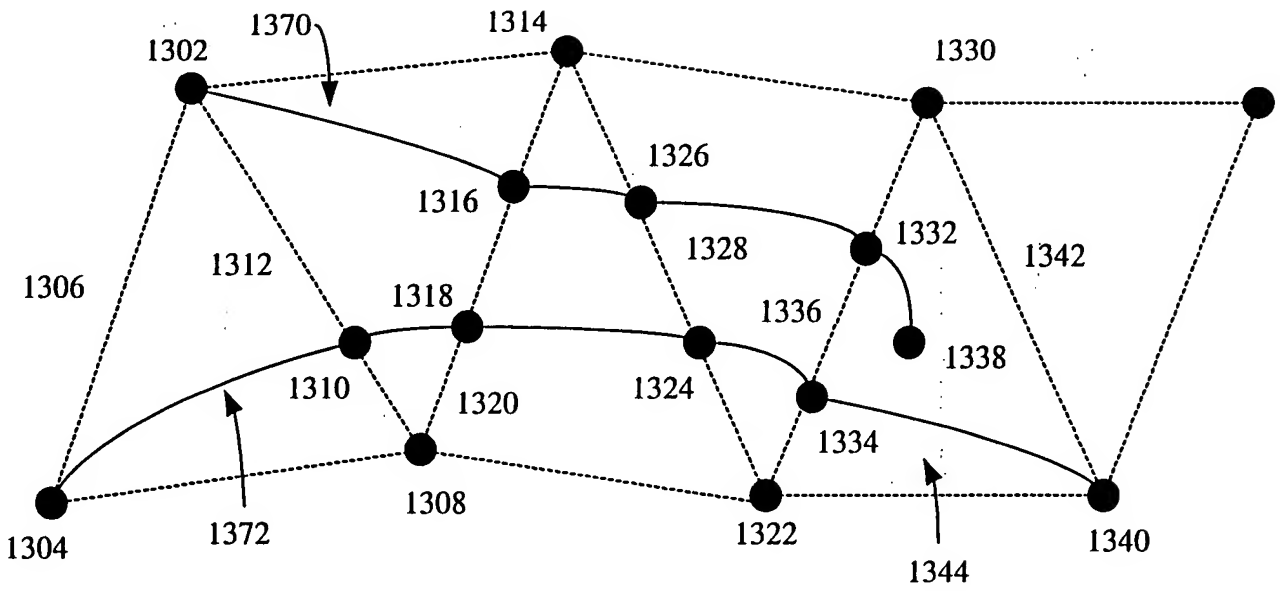
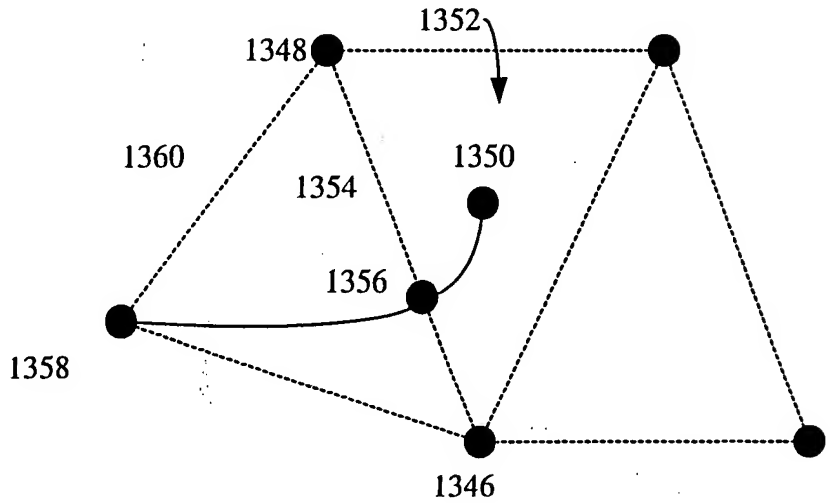


Figure 12

20TETO"h809901



Layer 2



Layer 3

Figure 13

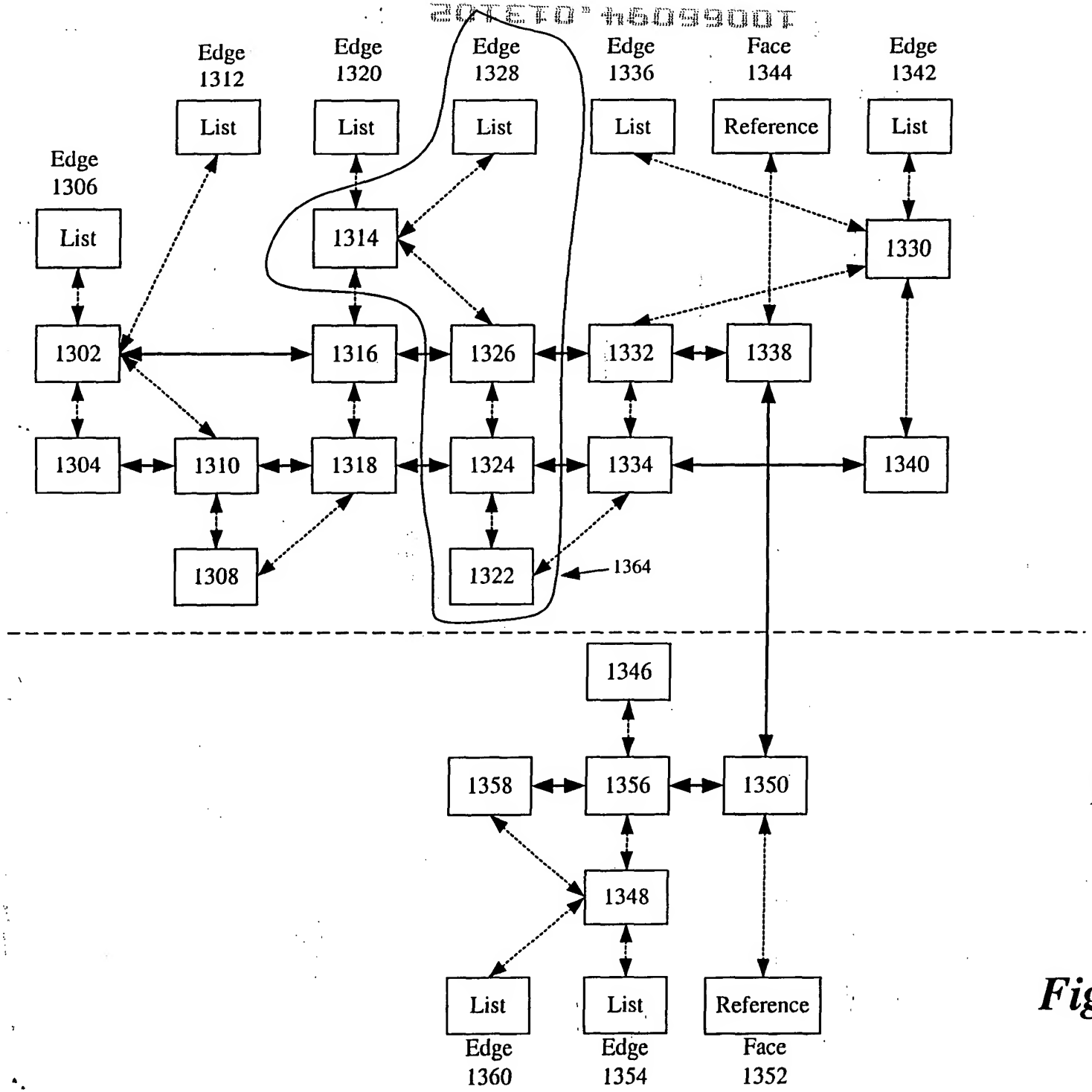

Edge 1312
Edge 1320
Edge 1328
Edge 1336
Face 1344
Edge 1342
Edge 1306
List
Reference
1314
1330
1302
1316
1326
1332
1338
1304
1310
1318
1324
1334
1340
1308
1322
1364
1346
1358
1356
1350
1348
Edge 1360
Edge 1354
Face 1352
Layer 2
Layer 3

Figure 14

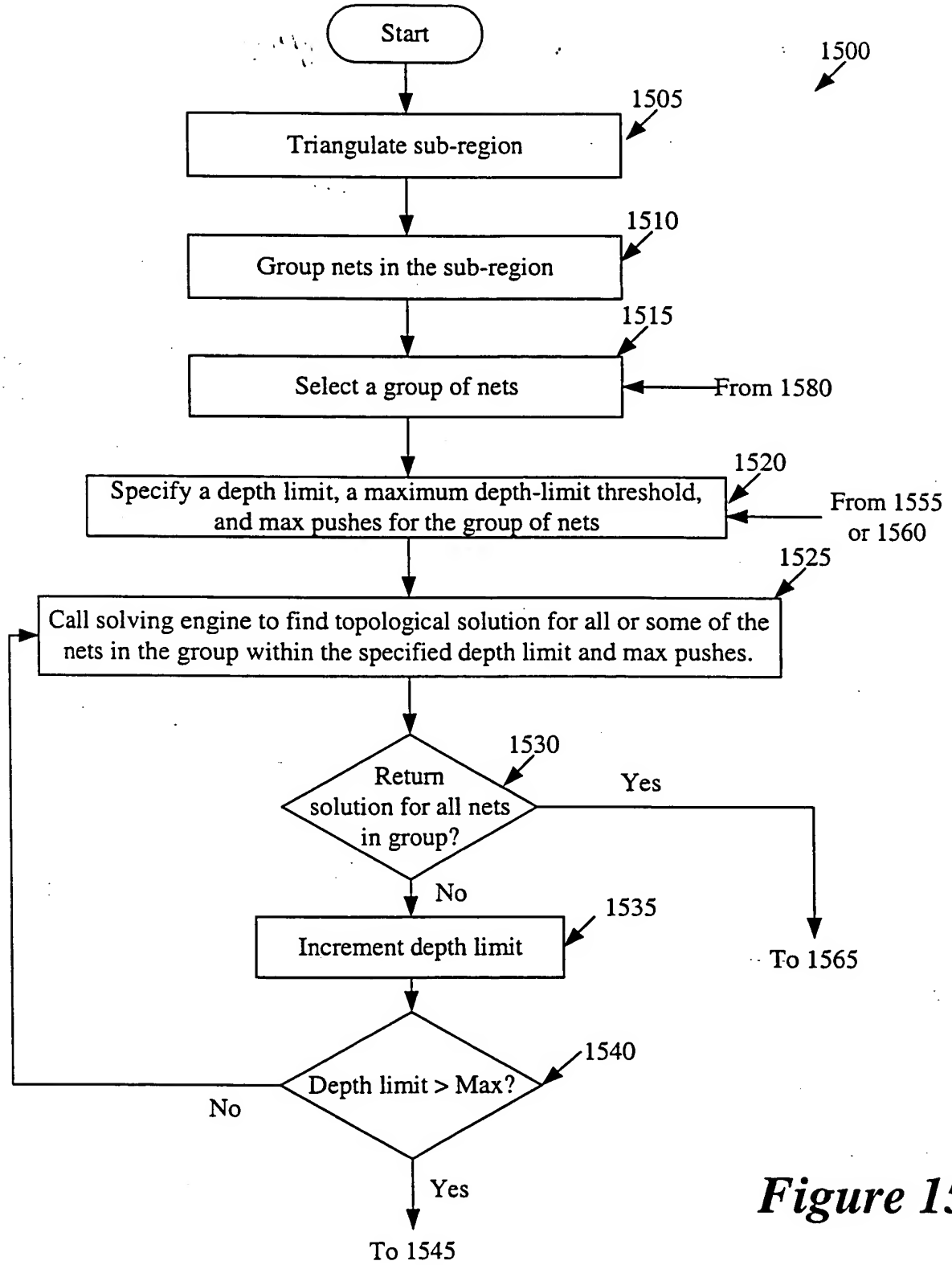


Figure 15A

Figure 15: Figure 15A / Figure 15B

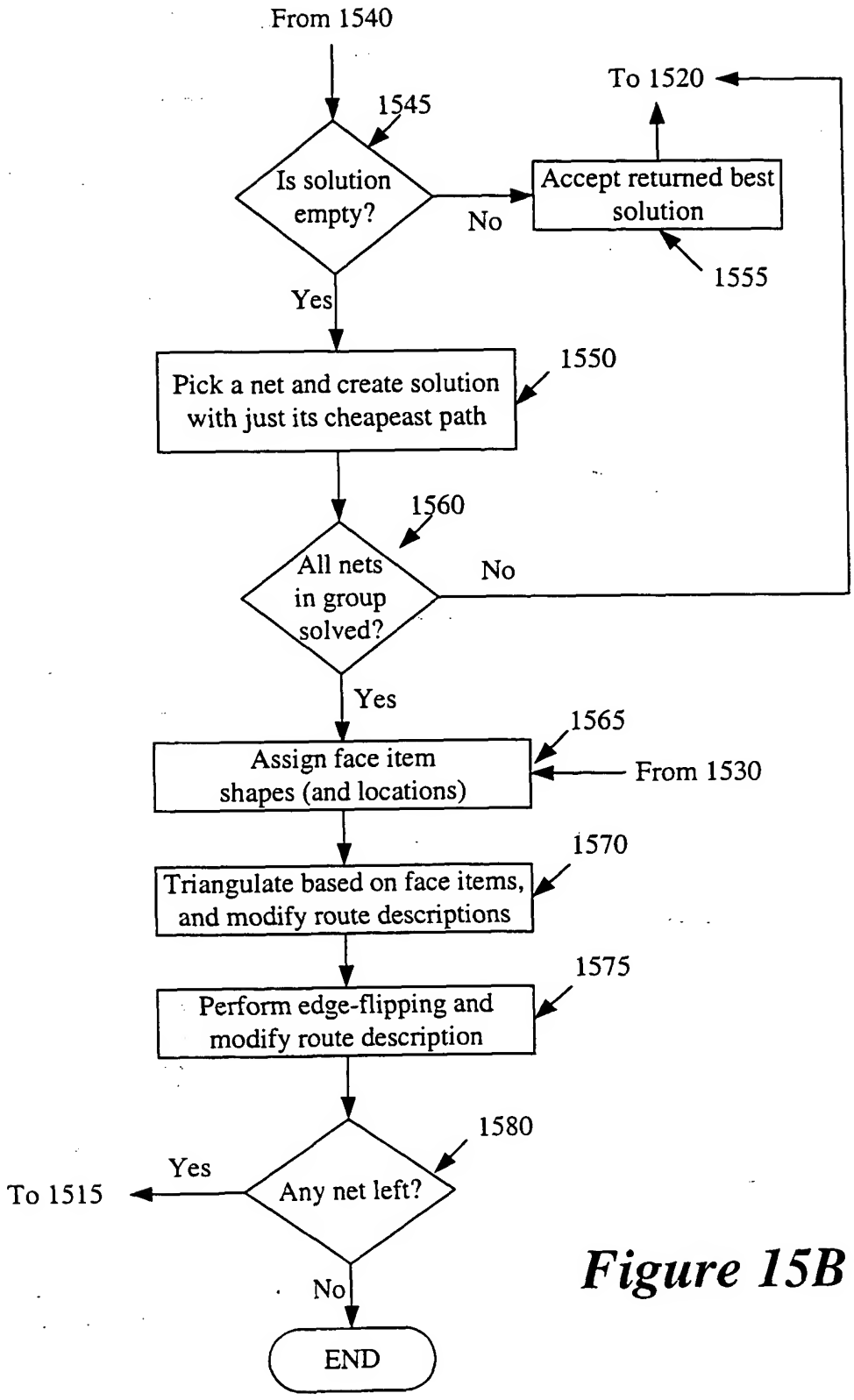

From 1540
1545
Is solution empty?
No
Accept returned best solution
To 1520
1555
Yes
Pick a net and create solution with just its cheapeast path
1550
1560
All nets in group solved?
No
Yes
1565
Assign face item shapes (and locations)
From 1530
1570
Triangulate based on face items, and modify route descriptions
1575
Perform edge-flipping and modify route description
1580
Yes
To 1515
Any net left?
No
END

Figure 15B

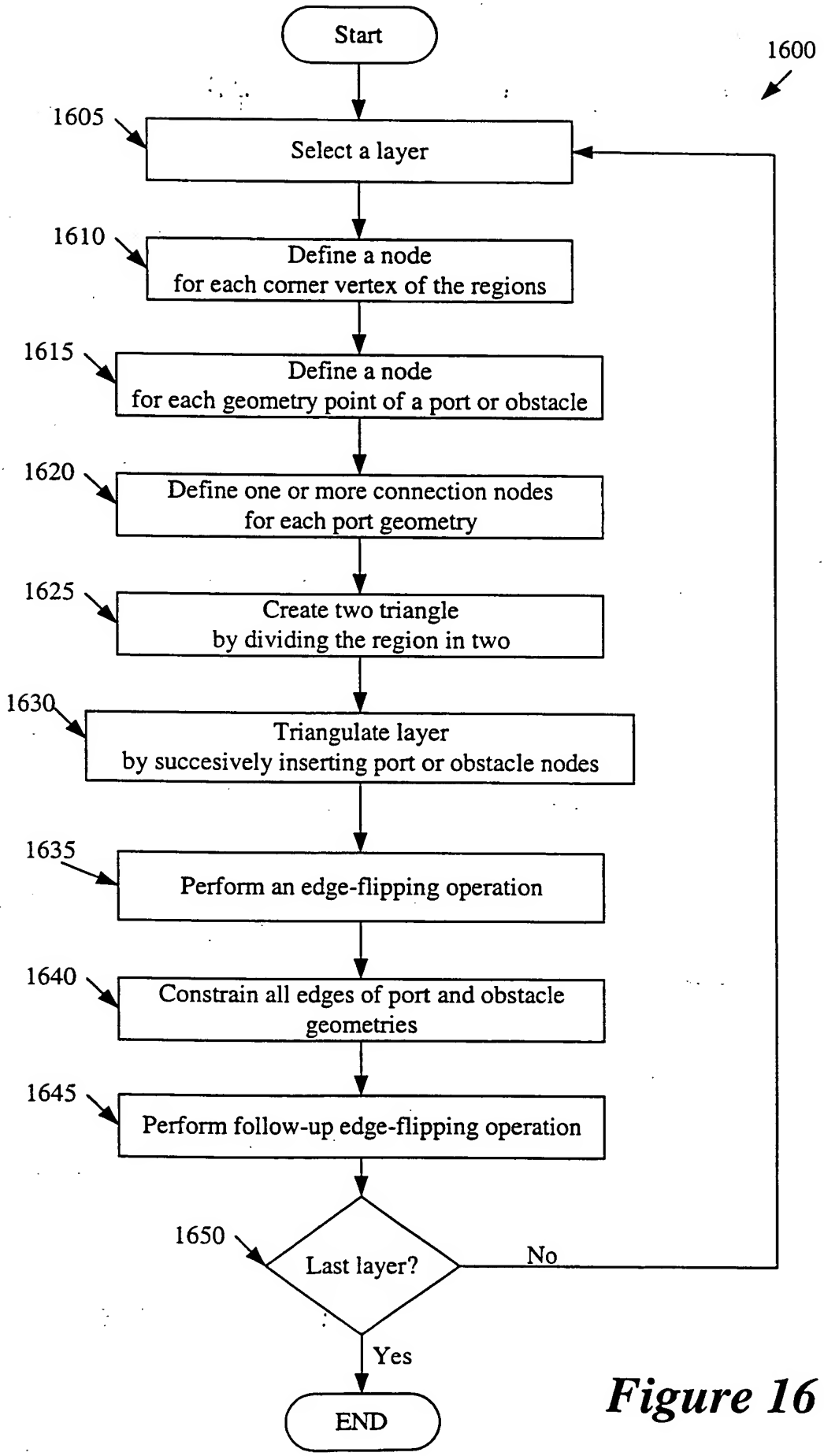

Start
1600
1605
Select a layer
1610
Define a node
for each corner vertex of the regions
1615
Define a node
for each geometry point of a port or obstacle
1620
Define one or more connection nodes
for each port geometry
1625
Create two triangle
by dividing the region in two
1630
Triangulate layer
by succesively inserting port or obstacle nodes
1635
Perform an edge-flipping operation
1640
Constrain all edges of port and obstacle
geometries
1645
Perform follow-up edge-flipping operation
1650
Last layer?
No
Yes
END

Figure 16

20EE0"A609900T

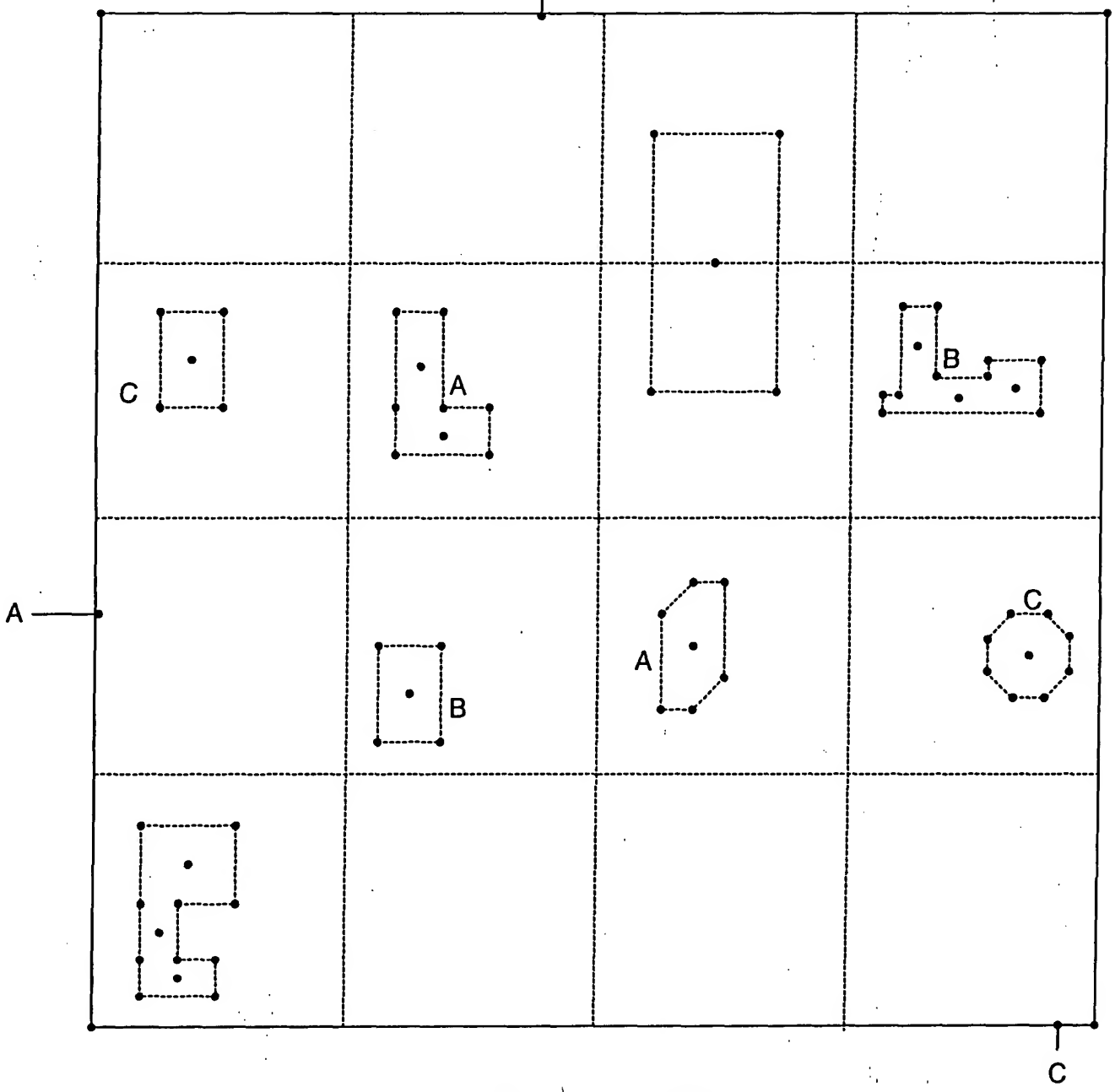


Figure 17

20TETO"h6099001

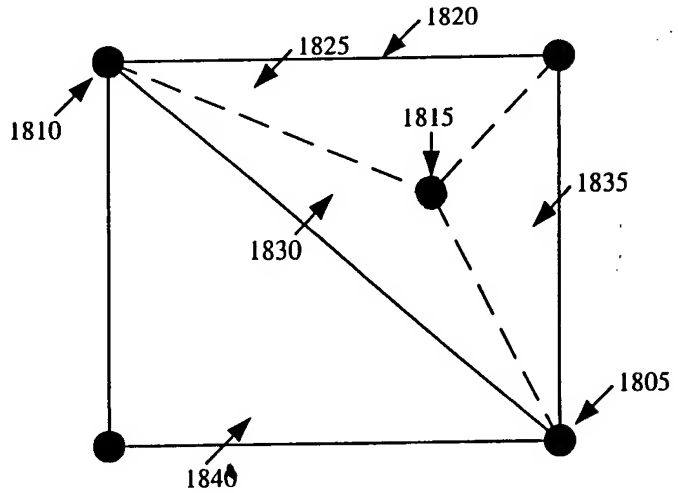


Figure 18

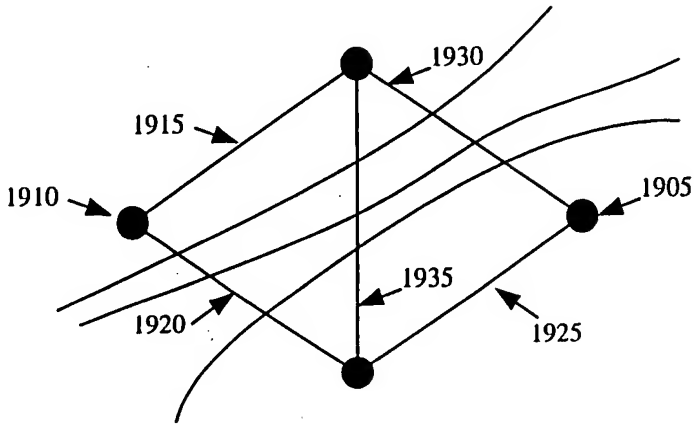


Figure 19

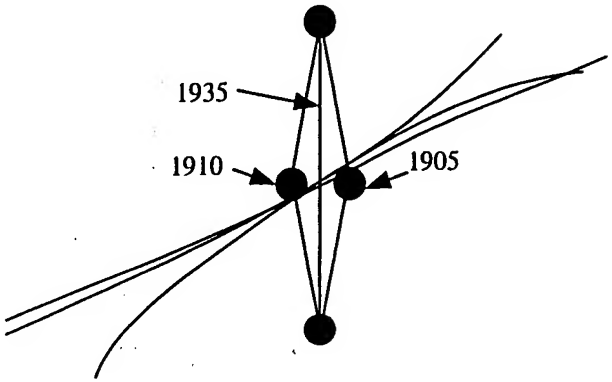


Figure 20

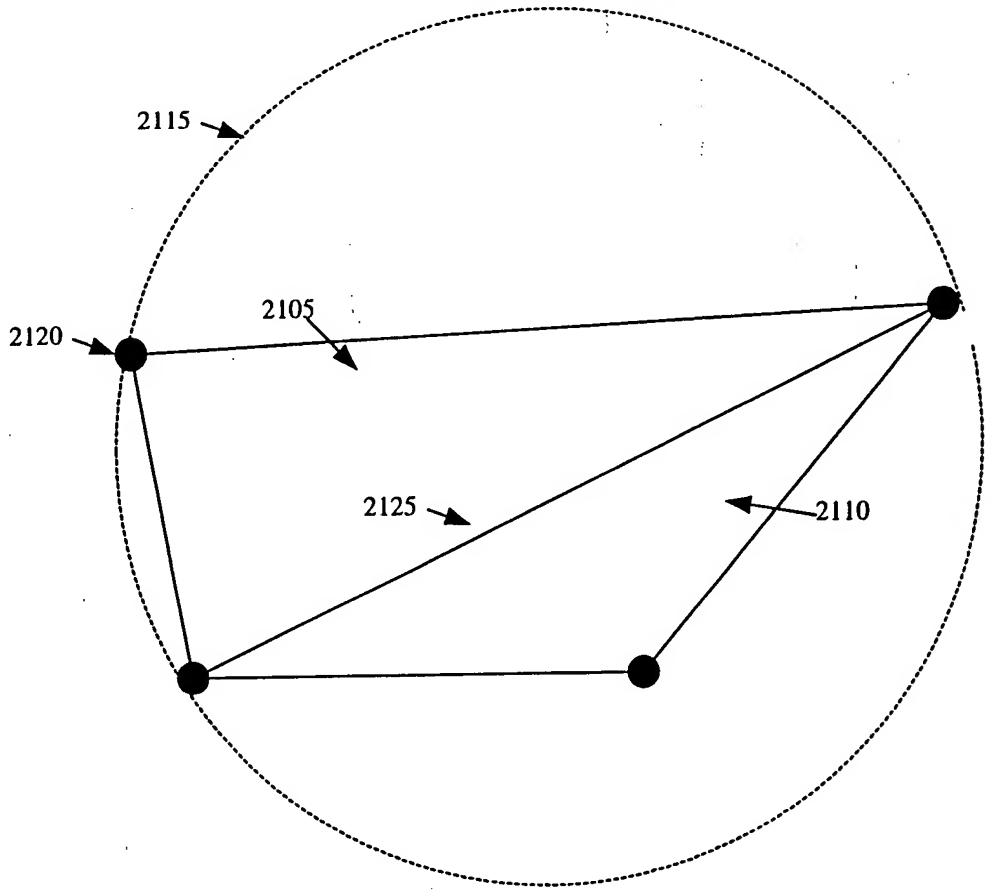


Figure 21

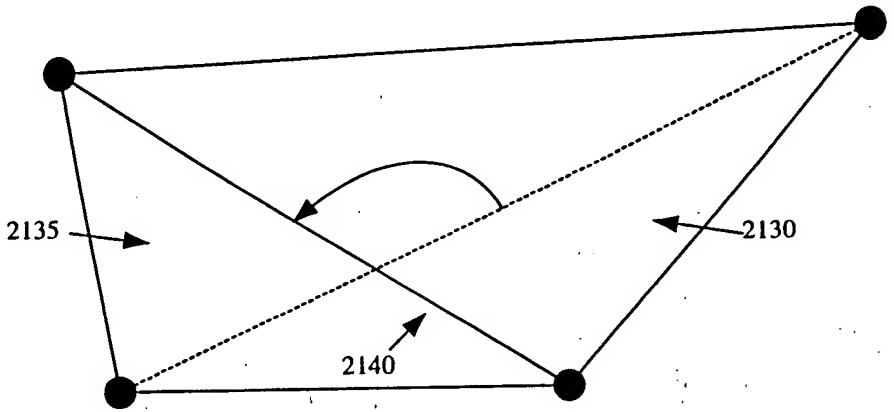


Figure 22

20TETO "t6099001

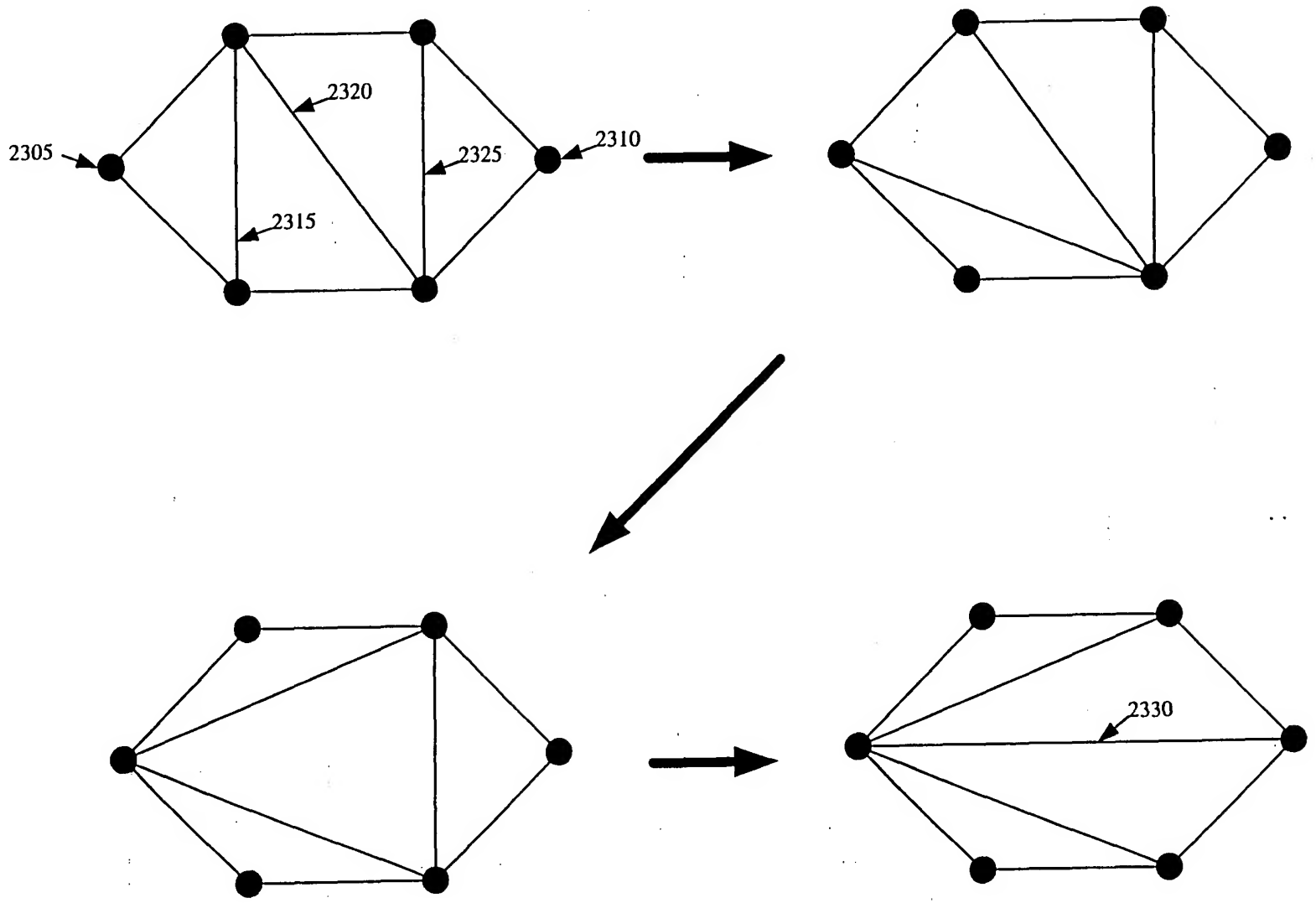


Figure 23

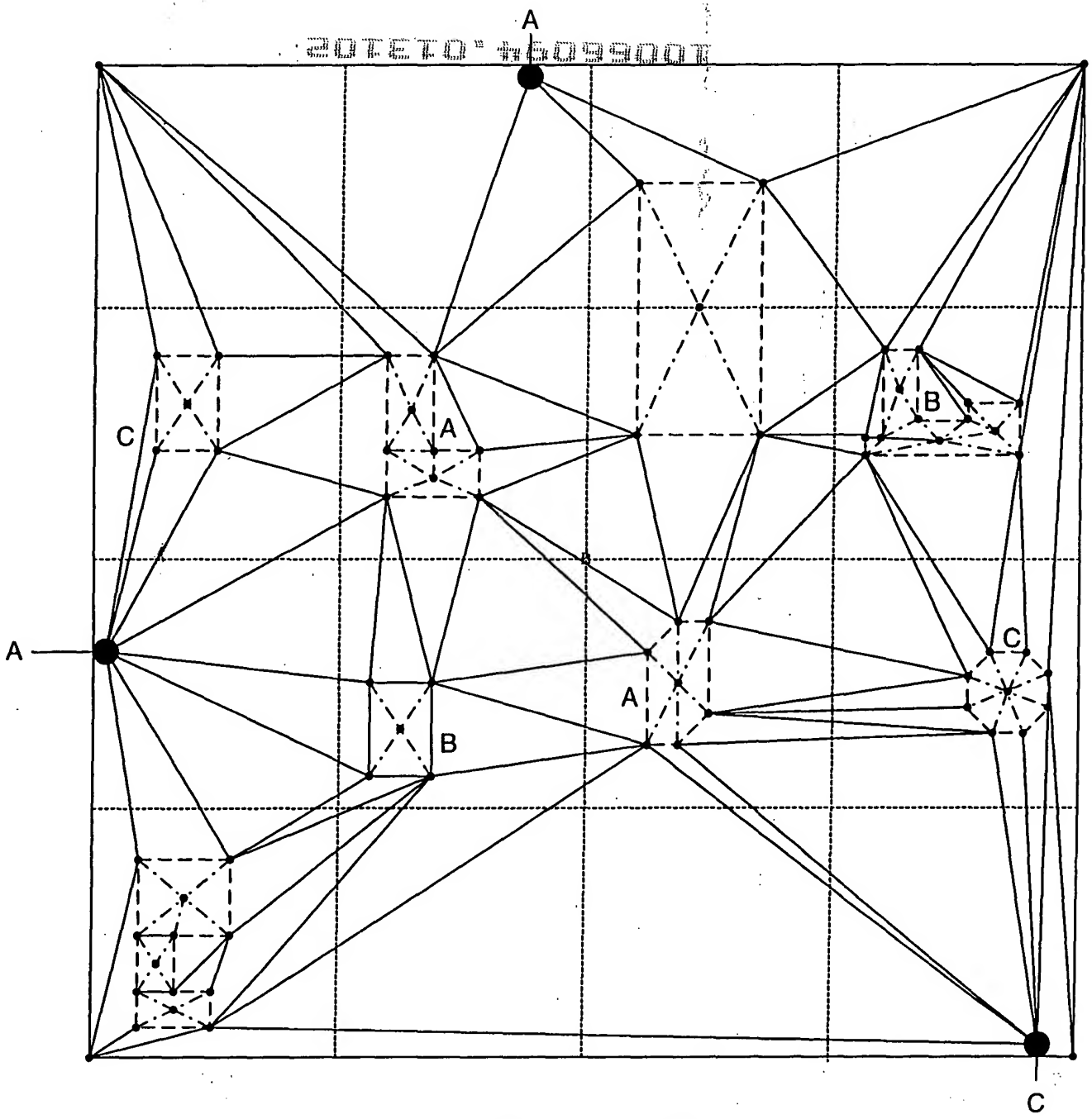


Figure 24

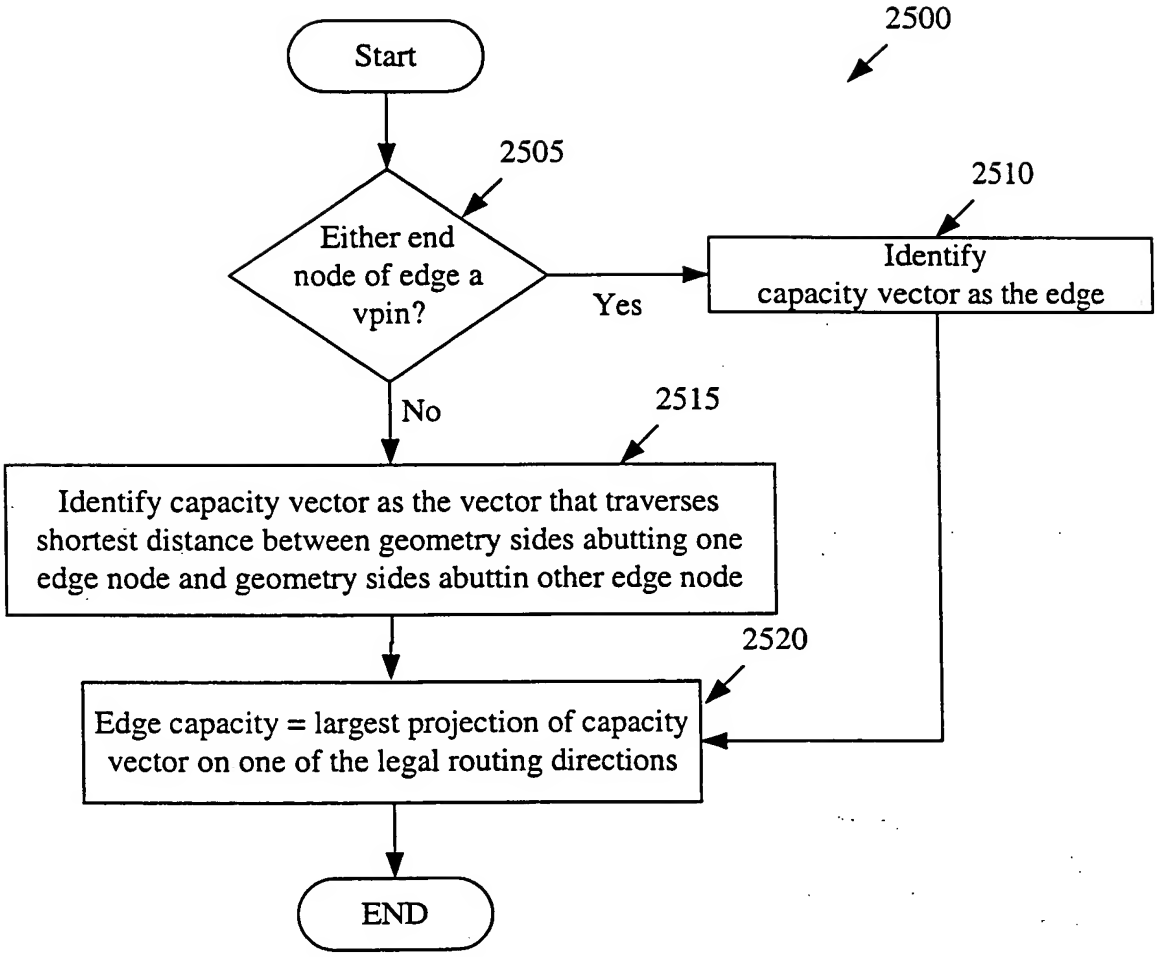

2500
Start
2505
Either end node of edge a vpin?
Yes
2510
Identify capacity vector as the edge
No
2515
Identify capacity vector as the vector that traverses shortest distance between geometry sides abutting one edge node and geometry sides abuttin other edge node
2520
Edge capacity = largest projection of capacity vector on one of the legal routing directions
END

Figure 25

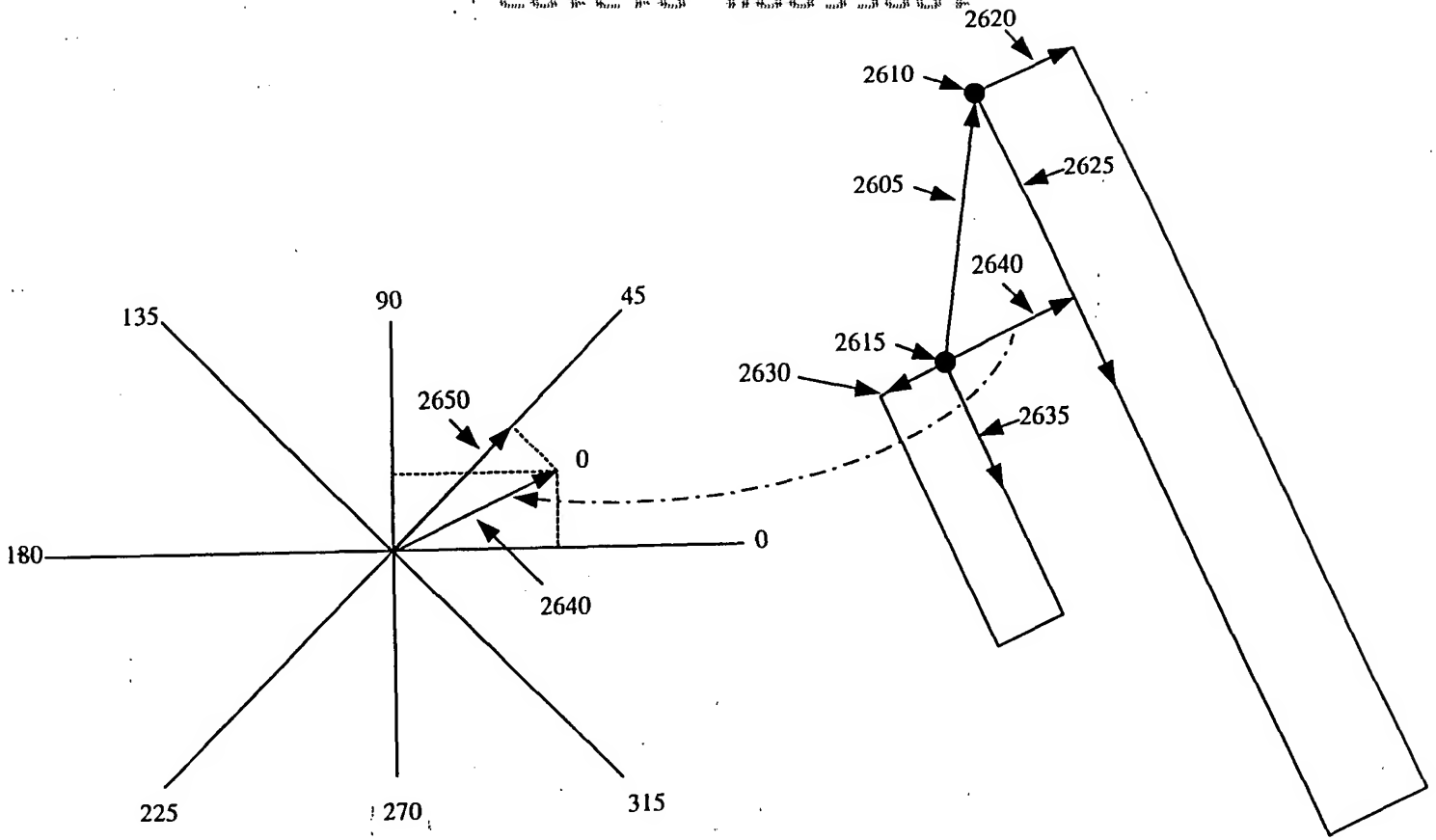

2620
2610
2625
2605
2640
2615
2630
2635
90
45
135
2650
0
180
0
2640
225
270
315

Figure 26

20TETO"h6099001

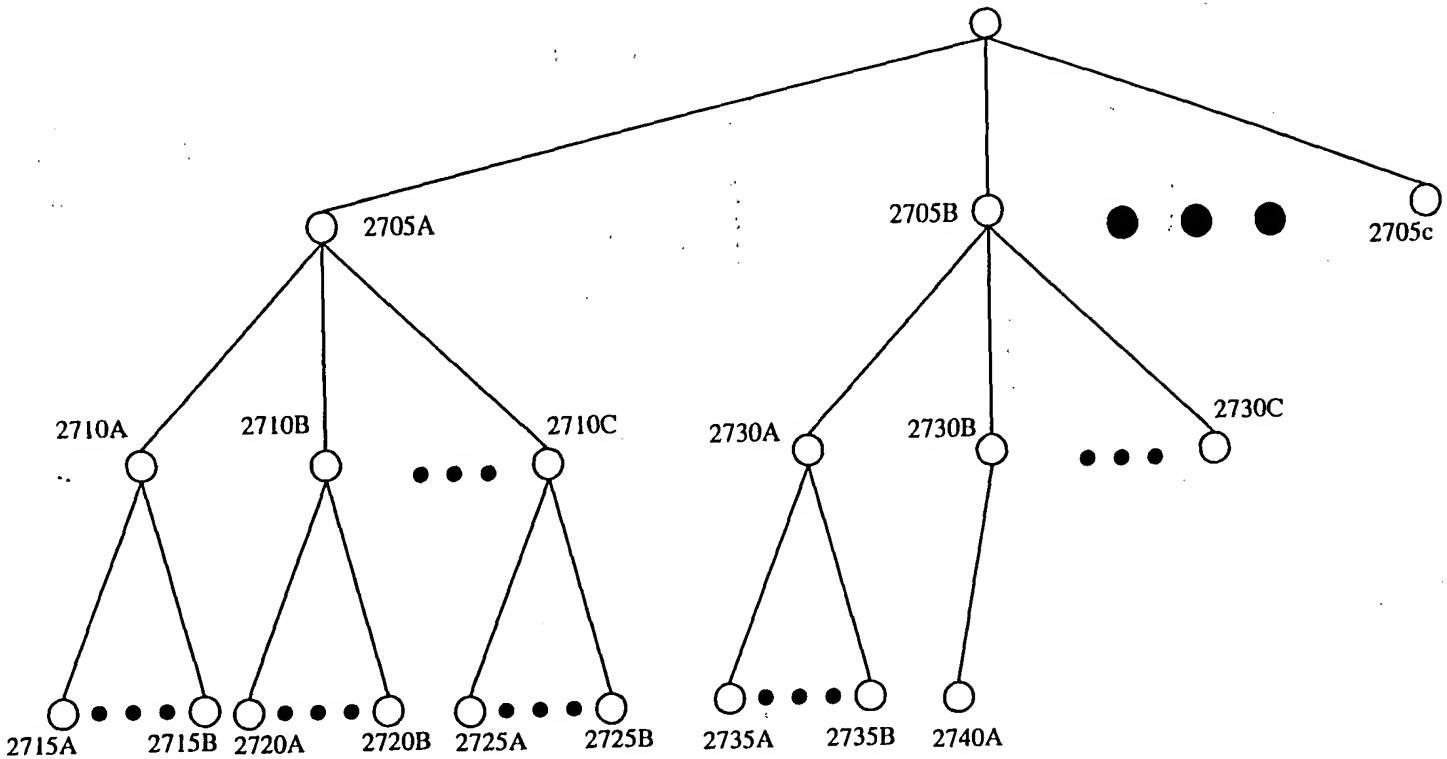


Figure 27

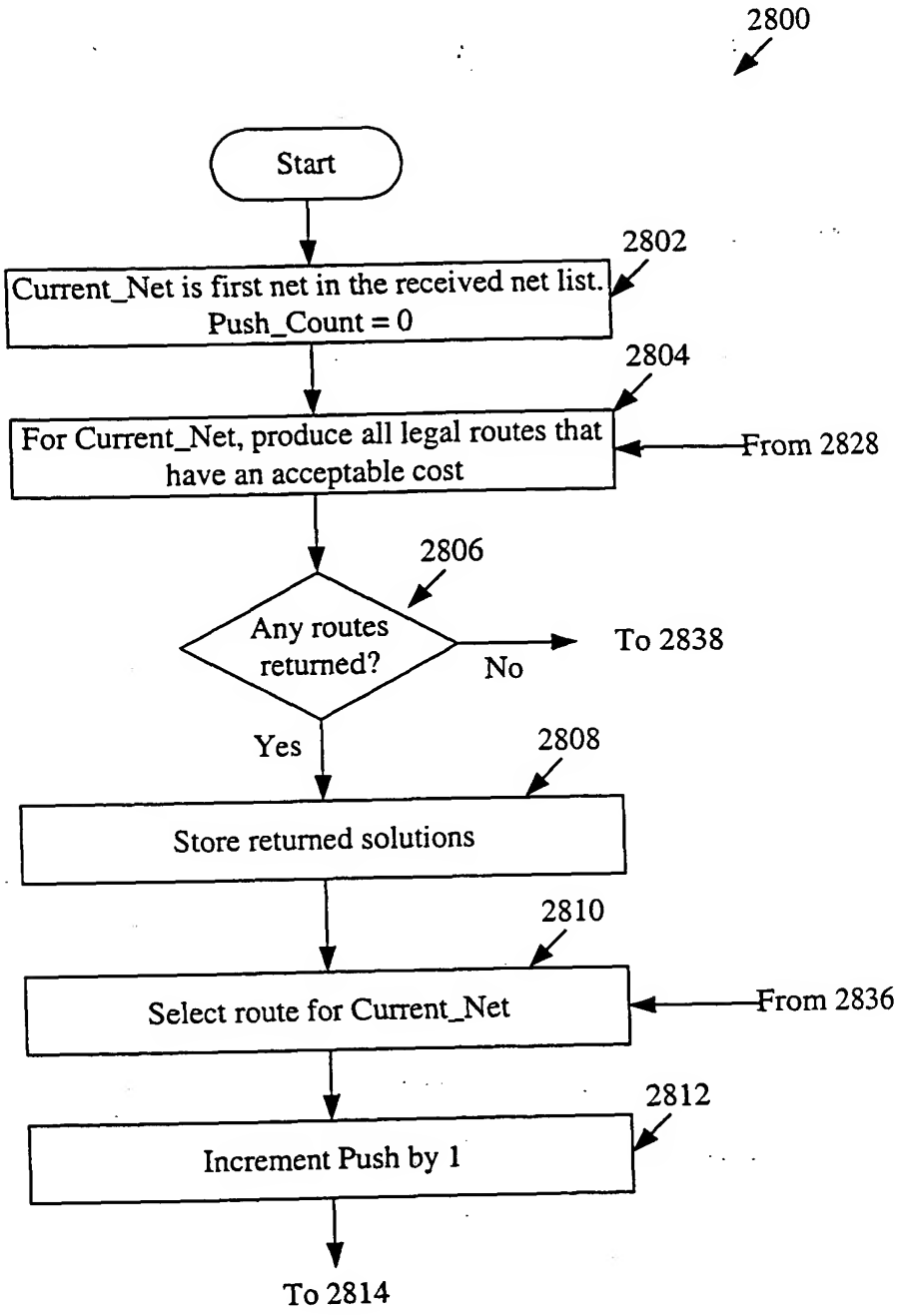


Figure 28A

Figure 28: ***Figure 28A / Figure 28B / Figure 28C***

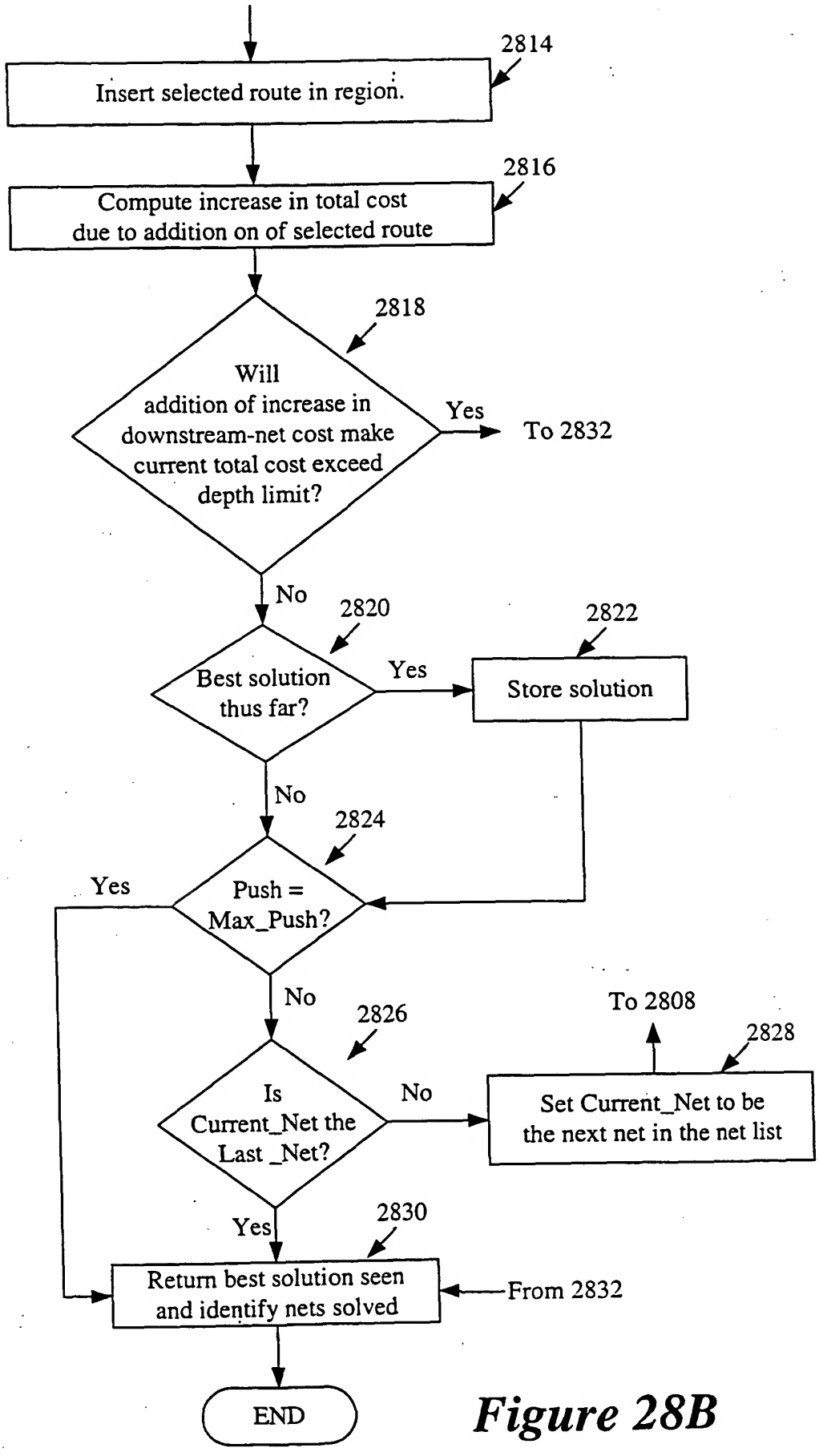

2814
Insert selected route in region.
2816
Compute increase in total cost due to addition on of selected route
2818
Will addition of increase in downstream-net cost make current total cost exceed depth limit?
Yes
To 2832
No
2820
Best solution thus far?
Yes
2822
Store solution
No
2824
Push = Max_Push?
Yes
No
2826
Is Current_Net the Last _Net?
No
To 2808
2828
Set Current_Net to be the next net in the net list
Yes
2830
Return best solution seen and identify nets solved
From 2832
END

Figure 28B

Docket No. 50T5-000104

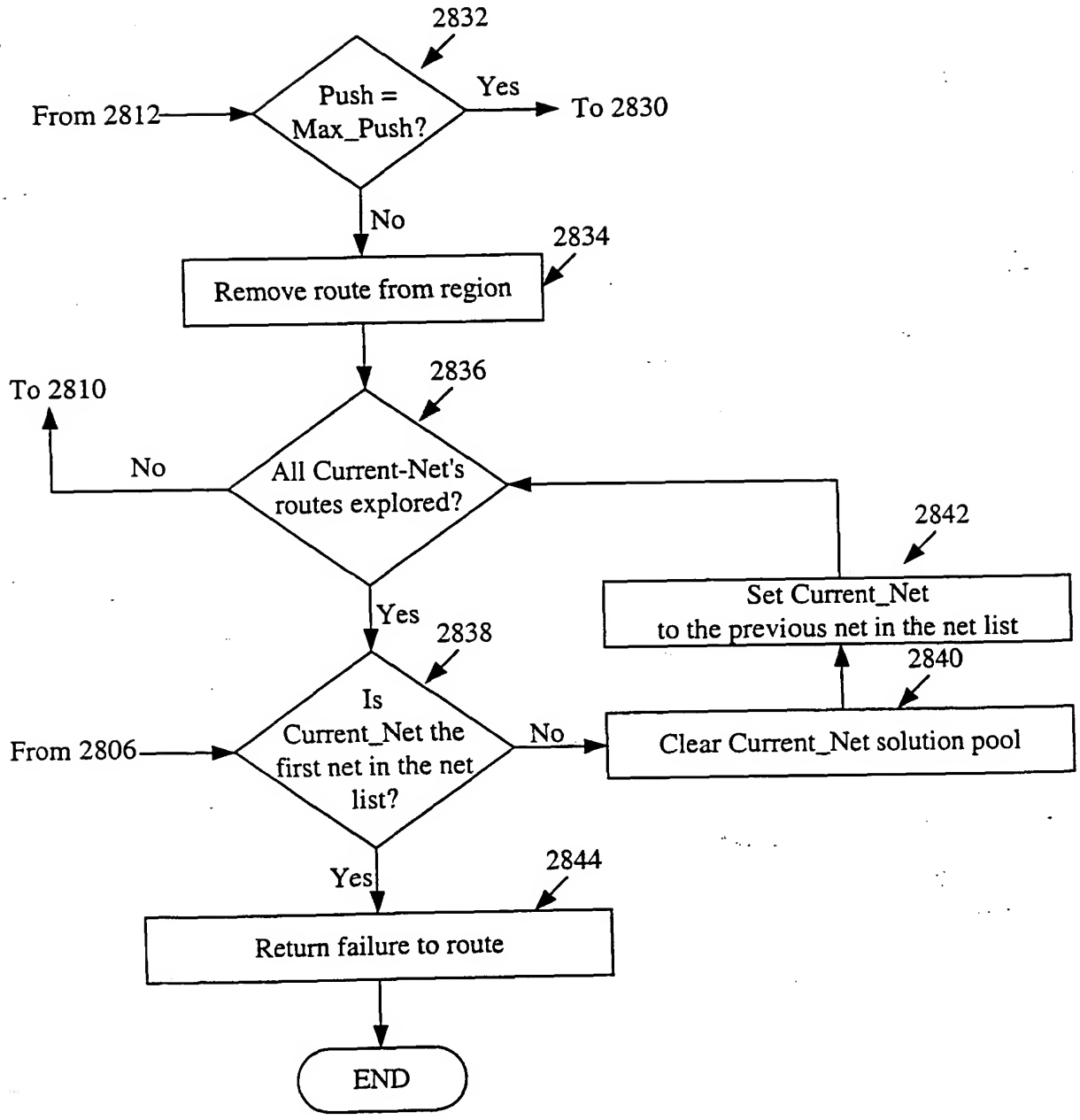


Figure 28C

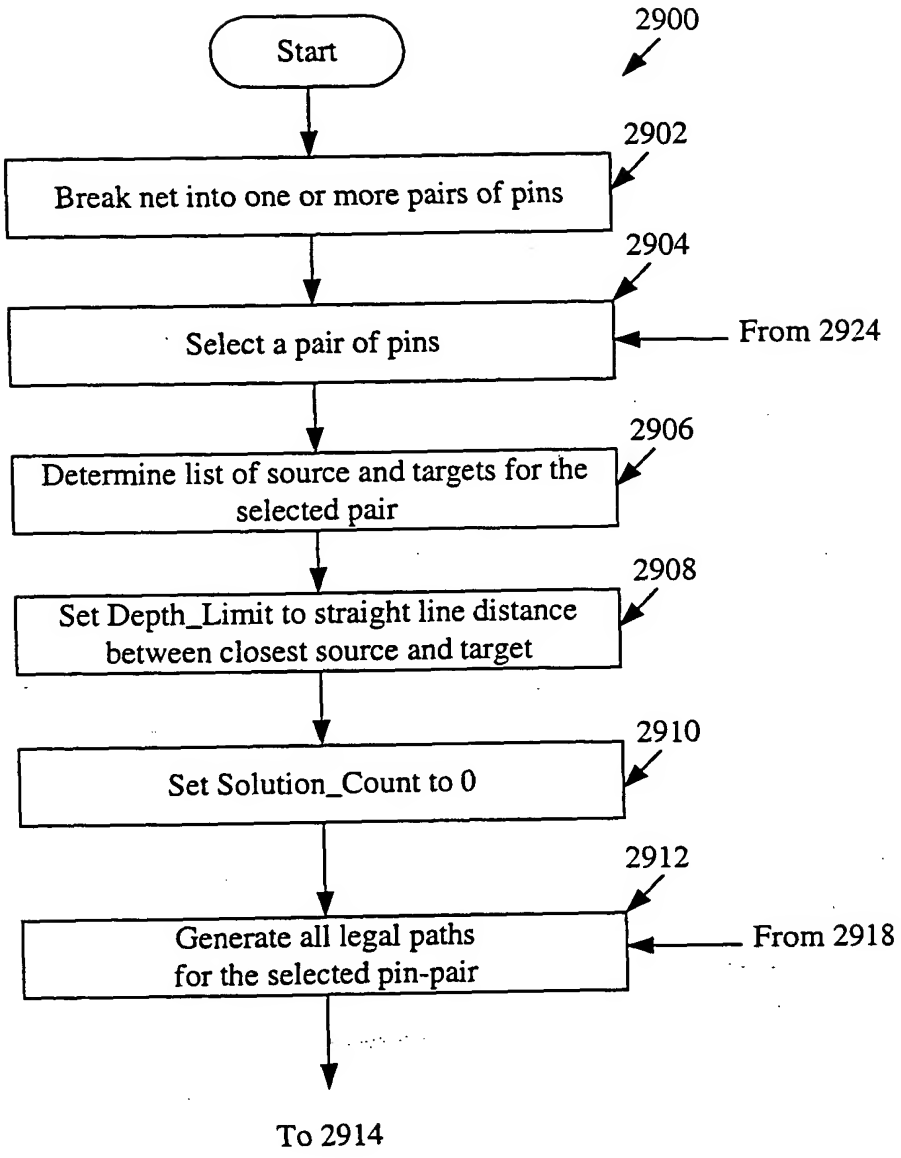


Figure 29A

Figure 29: Figure 29A / Figure 29B

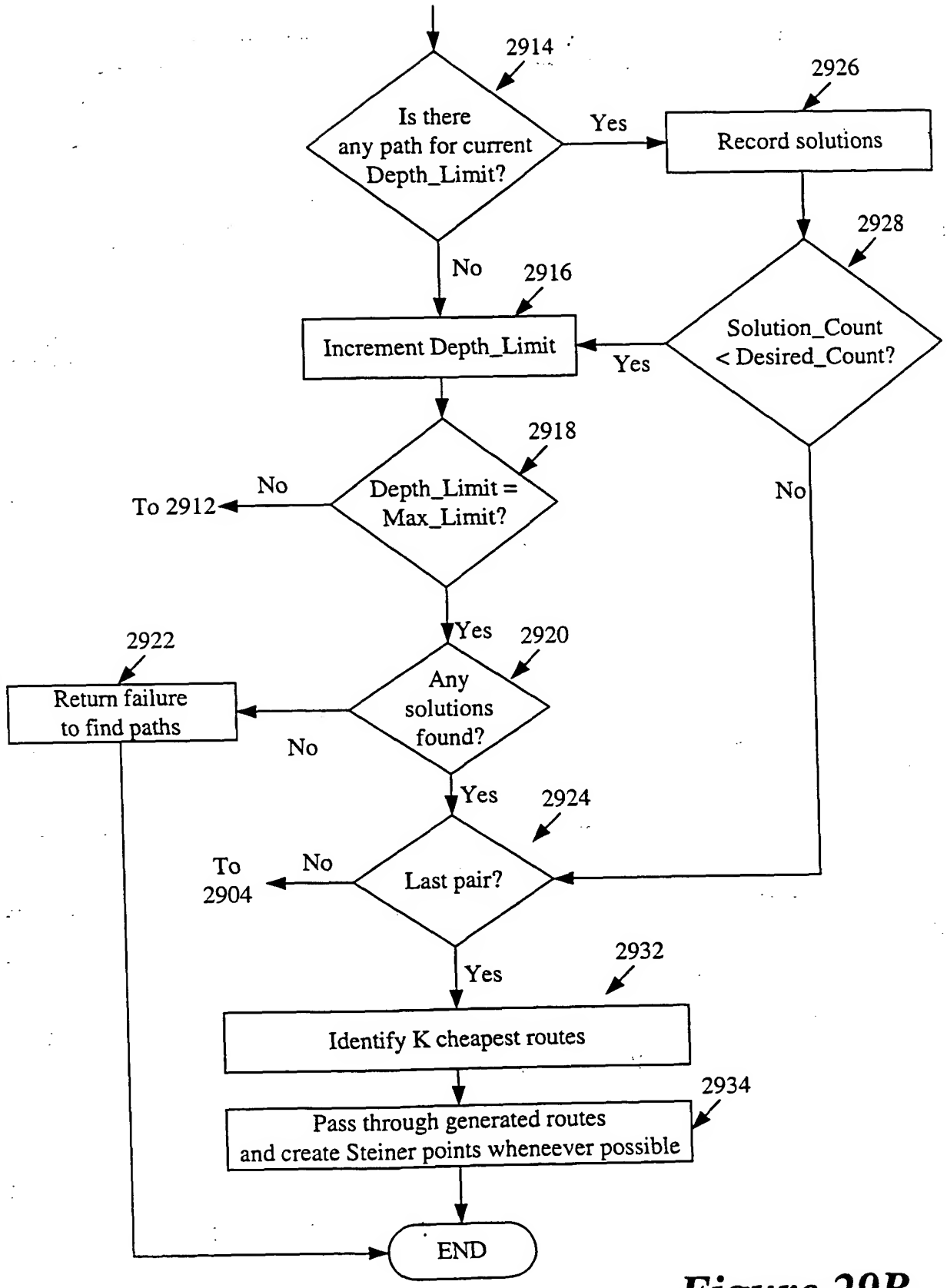

2914
Is there any path for current Depth_Limit?
Yes
2926
Record solutions
2928
No
2916
Solution_Count < Desired_Count?
Increment Depth_Limit
Yes
2918
No
To 2912
Depth_Limit = Max_Limit?
No
Yes
2920
2922
Any solutions found?
Return failure to find paths
No
Yes
2924
To 2904
No
Last pair?
Yes
2932
Identify K cheapest routes
2934
Pass through generated routes and create Steiner points whenever possible
END

Figure 29B

20TETO" h6099001

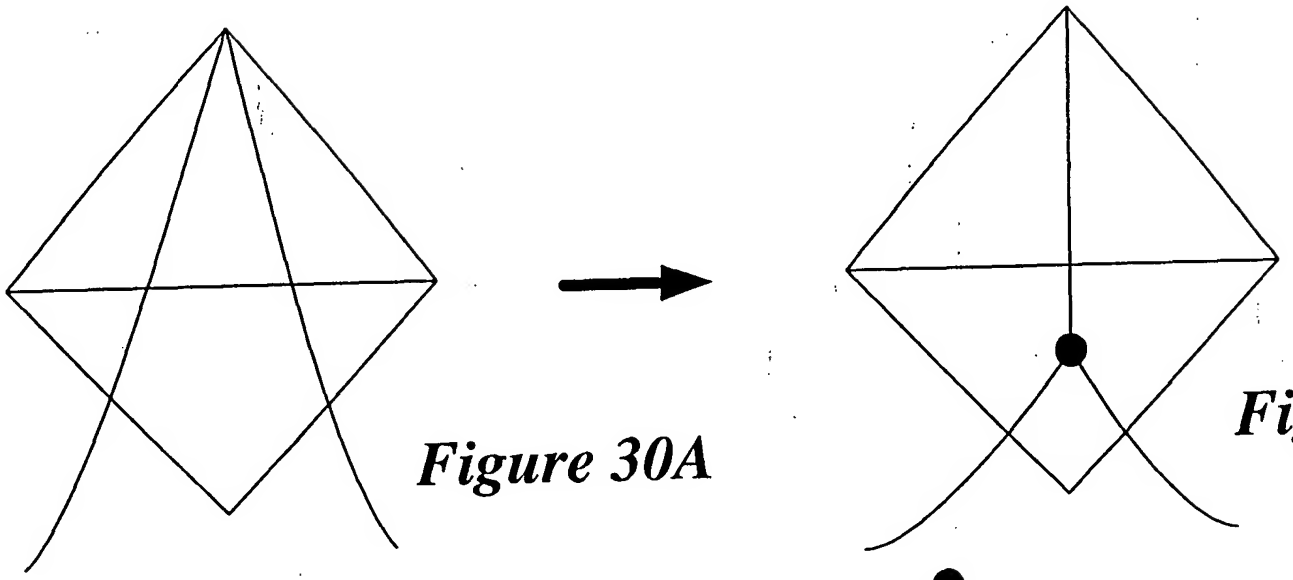


Figure 30A

Figure 30B

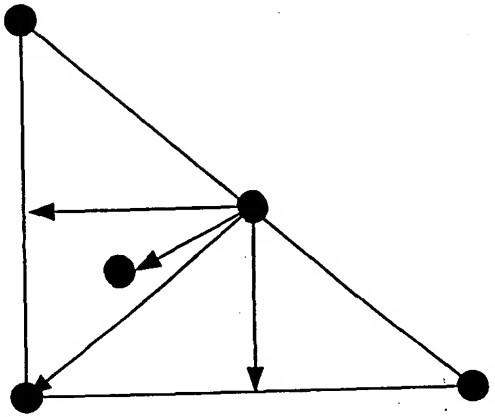


Figure 32

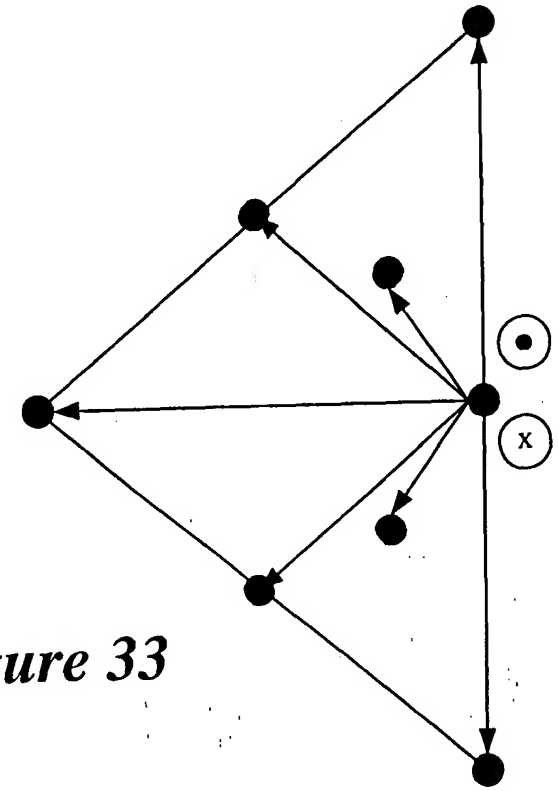


Figure 33

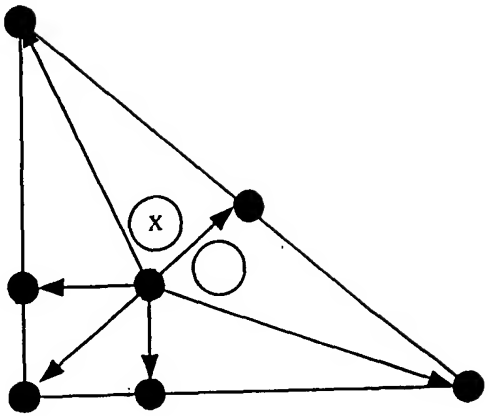


Figure 34

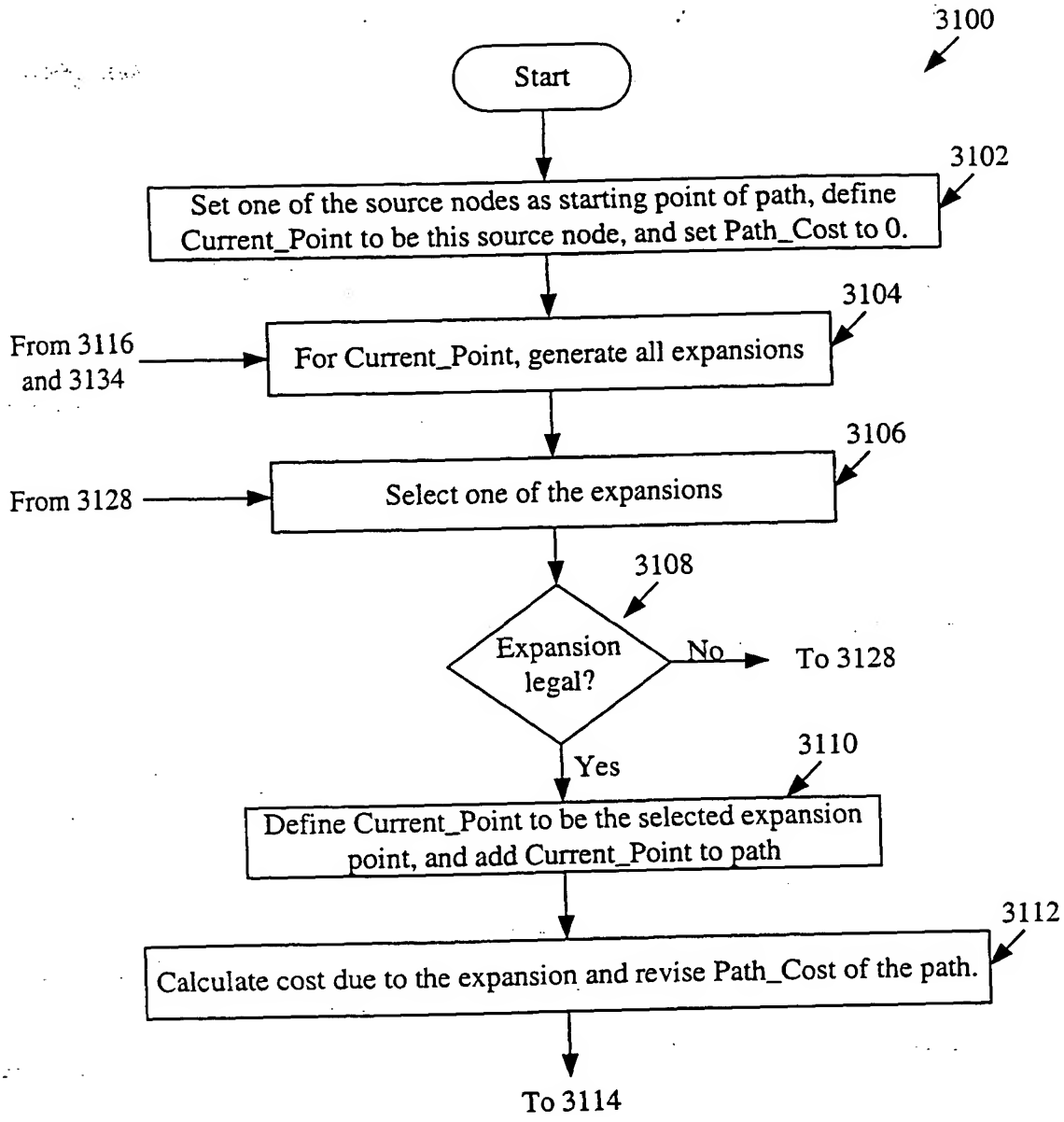


Figure 31A

Figure 31: Figure 31A / Figure 31B

20150304-0005E1

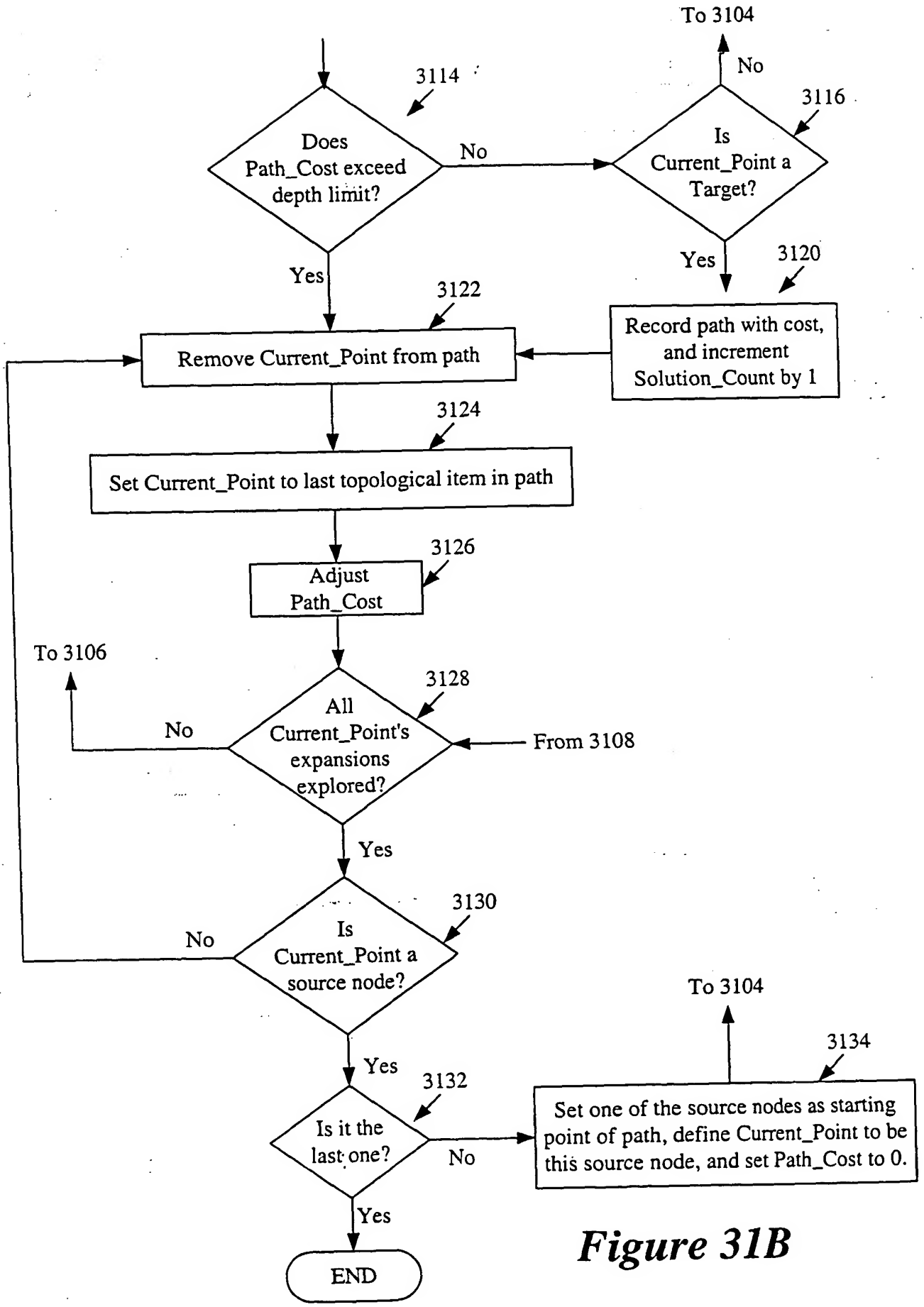


Figure 31B

20TETO"h6099001

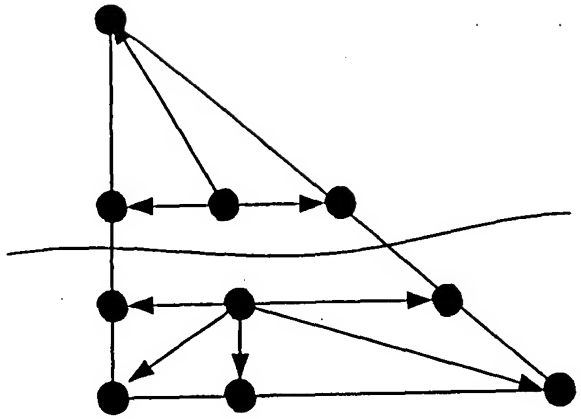


Figure 35

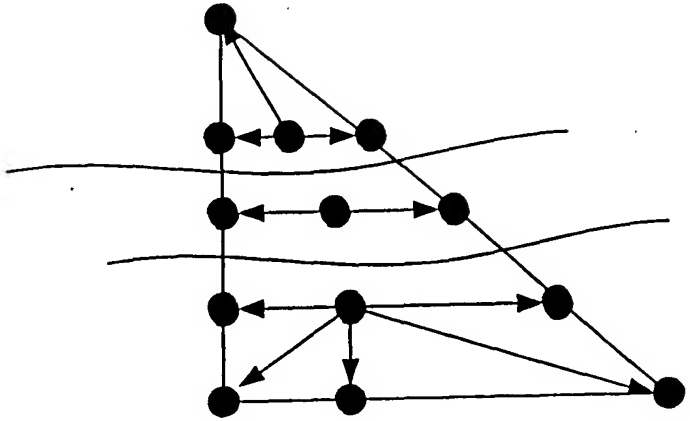


Figure 36

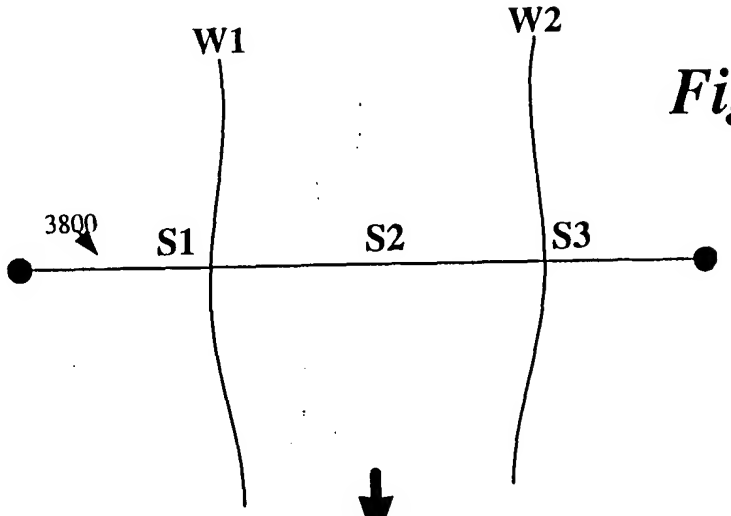


Figure 38A

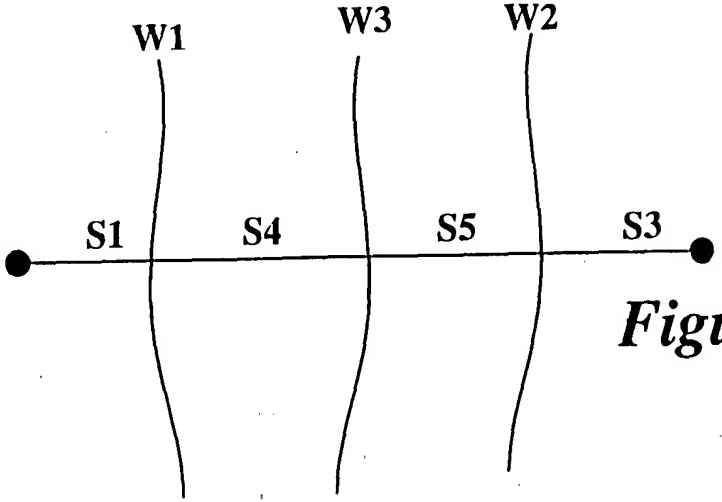


Figure 38B

20TETO"th099001

From: \ To:	Node	Face Item	Edge Item
Node	• Planarity • Vias	• Vias	• Planarity • Vias • Edge Capacity
Face Item	• Vias	• Vias	• Vias • Edge Capacity
Edge Item	• Planarity • Vias	• Vias	• Planarity • Vias • Edge Capacity

Figure 37

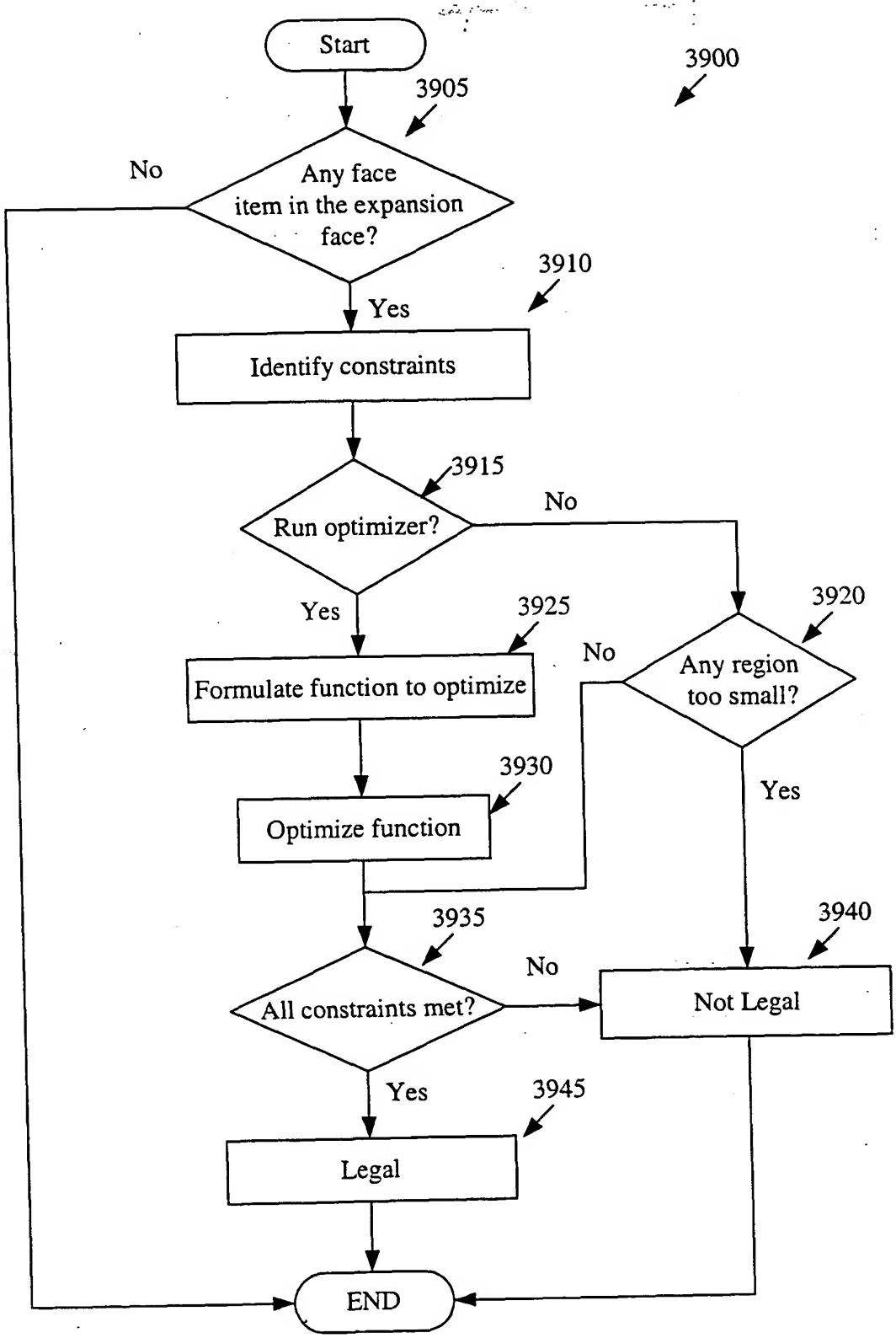

Start
3900
3905
Any face item in the expansion face?
No
Yes
3910
Identify constraints
3915
Run optimizer?
No
Yes
3925
Formulate function to optimize
3920
Any region too small?
No
Yes
3930
Optimize function
3935
All constraints met?
No
3940
Not Legal
Yes
3945
Legal
END

Figure 39A

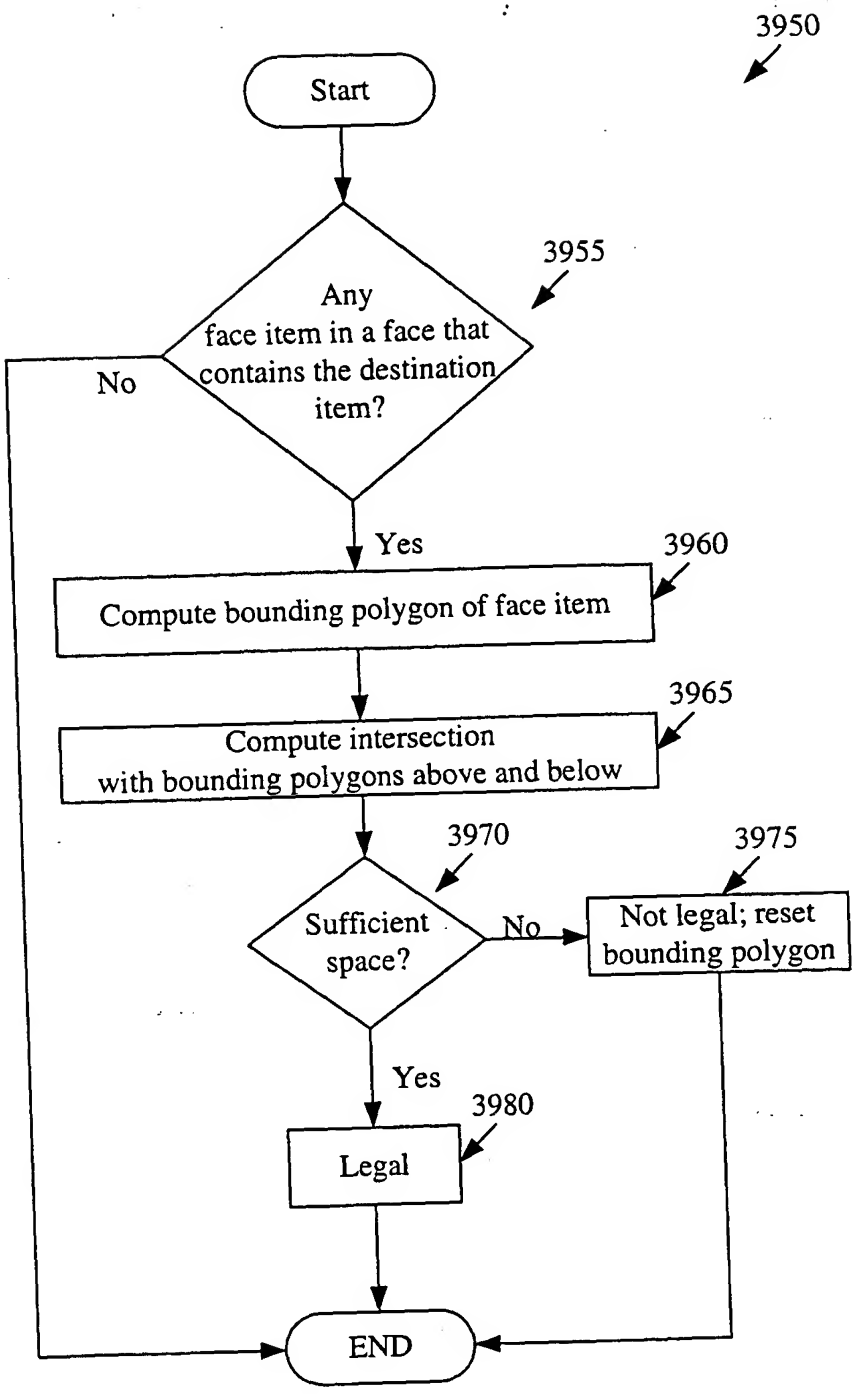

3950
Start
3955
Any face item in a face that contains the destination item?
No
Yes
3960
Compute bounding polygon of face item
3965
Compute intersection with bounding polygons above and below
3970
Sufficient space?
No
3975
Not legal; reset bounding polygon
Yes
3980
Legal
END

Figure 39B

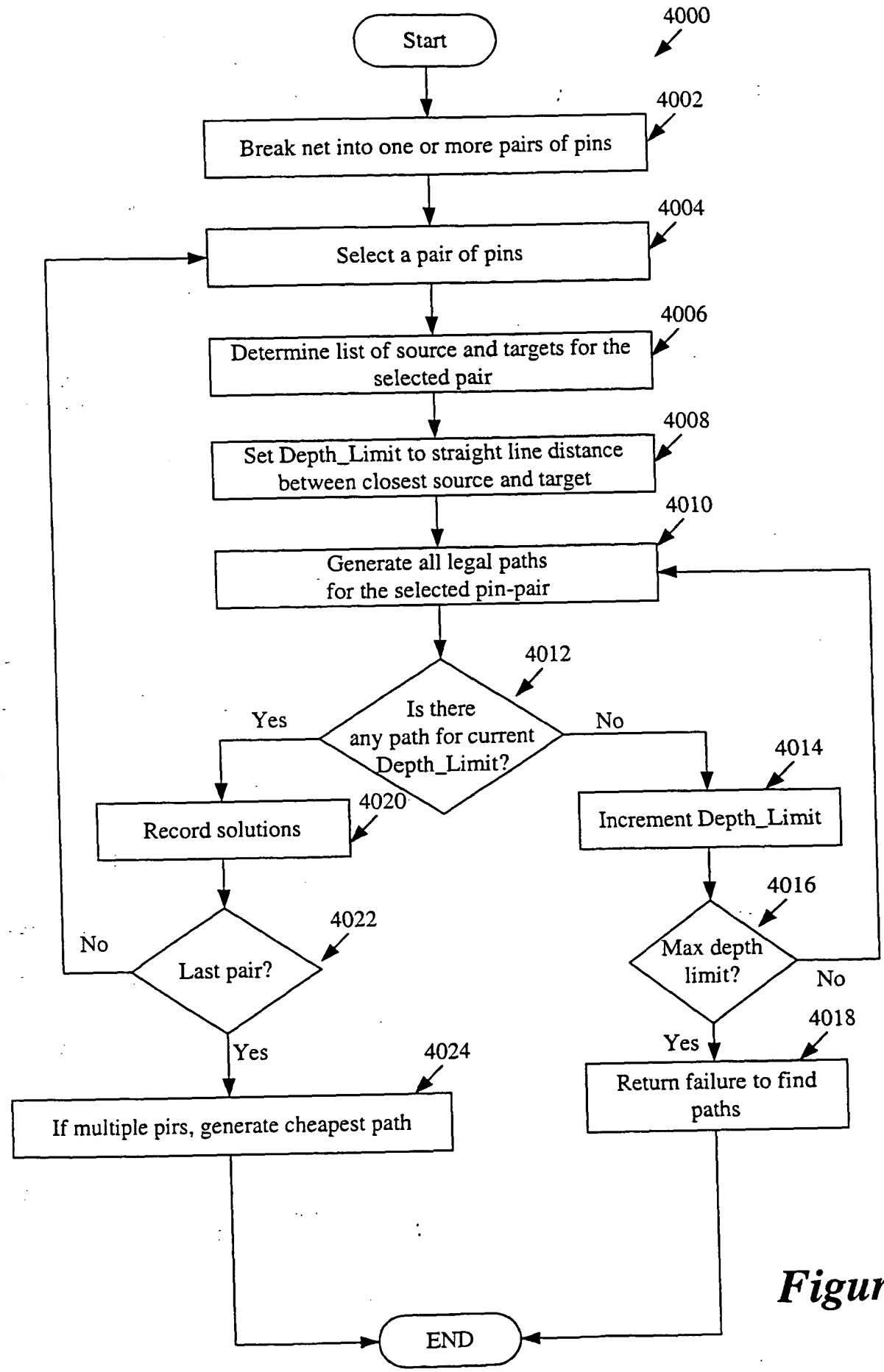

4000
Start
4002
Break net into one or more pairs of pins
4004
Select a pair of pins
4006
Determine list of source and targets for the selected pair
4008
Set Depth_Limit to straight line distance between closest source and target
4010
Generate all legal paths for the selected pin-pair
4012
Is there any path for current Depth_Limit?
Yes
No
4020
Record solutions
4014
Increment Depth_Limit
4016
Max depth limit?
No
Yes
4018
Return failure to find paths
4022
Last pair?
No
Yes
4024
If multiple pirs, generate cheapest path
END

Figure 40

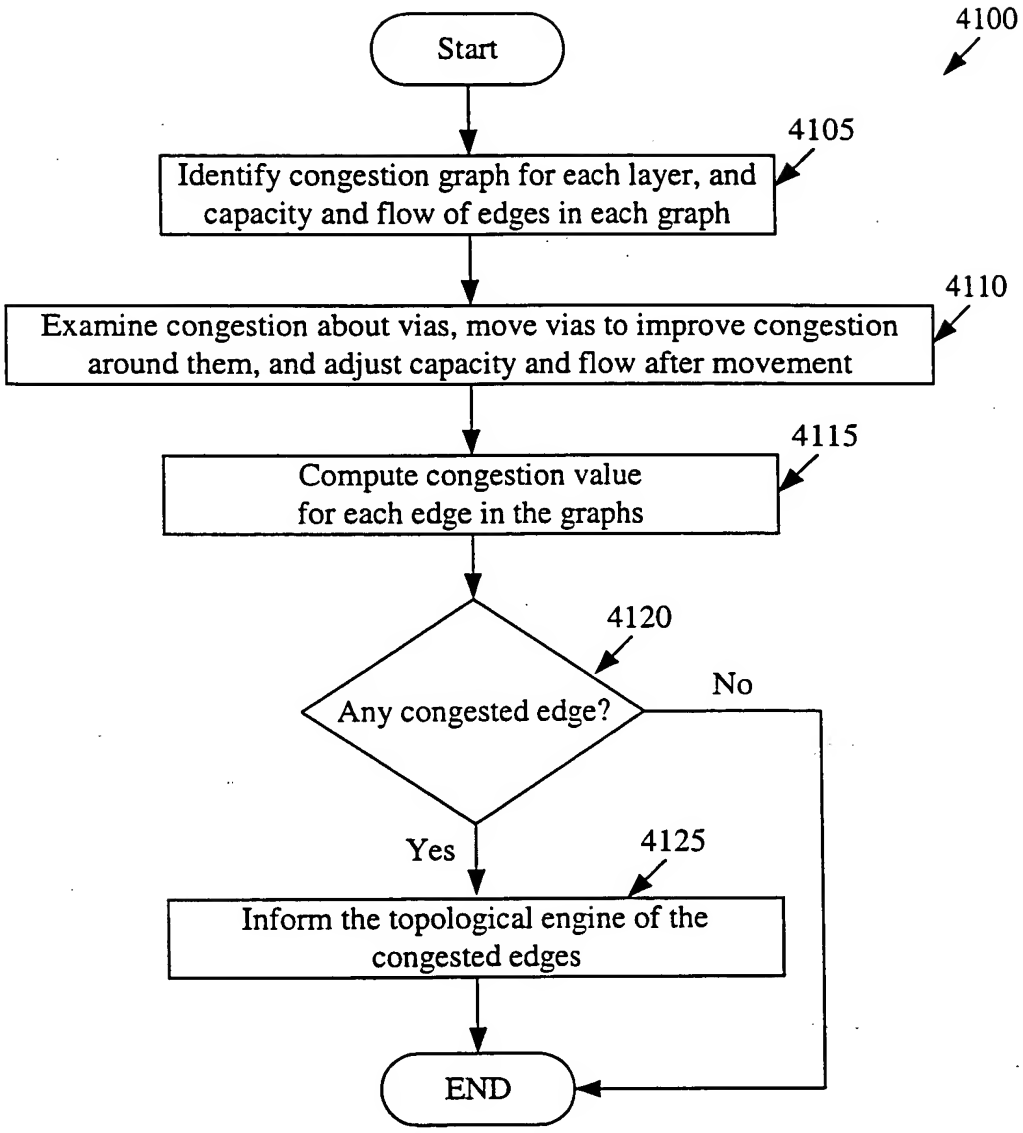

4100
Start
4105
Identify congestion graph for each layer, and capacity and flow of edges in each graph
4110
Examine congestion about vias, move vias to improve congestion around them, and adjust capacity and flow after movement
4115
Compute congestion value for each edge in the graphs
4120
Any congested edge?
No
Yes
4125
Inform the topological engine of the congested edges
END

Figure 41

20TETO"4609901

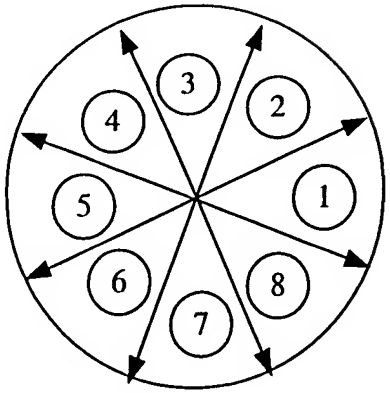


Figure 42

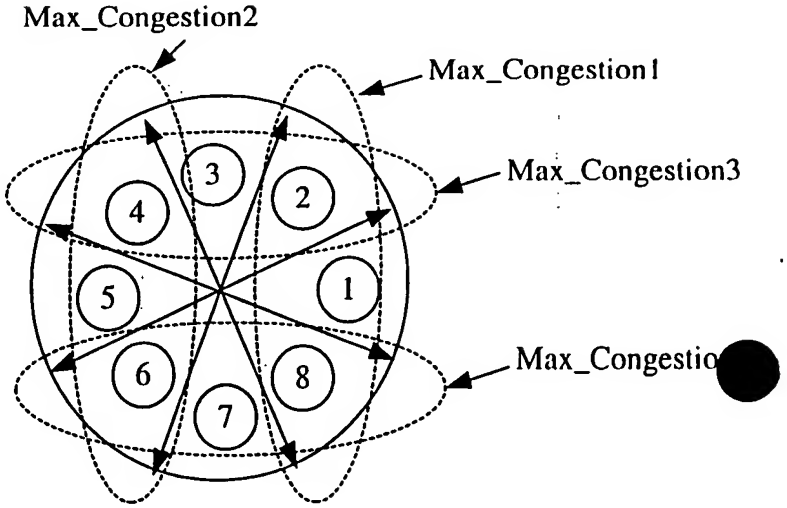


Figure 44

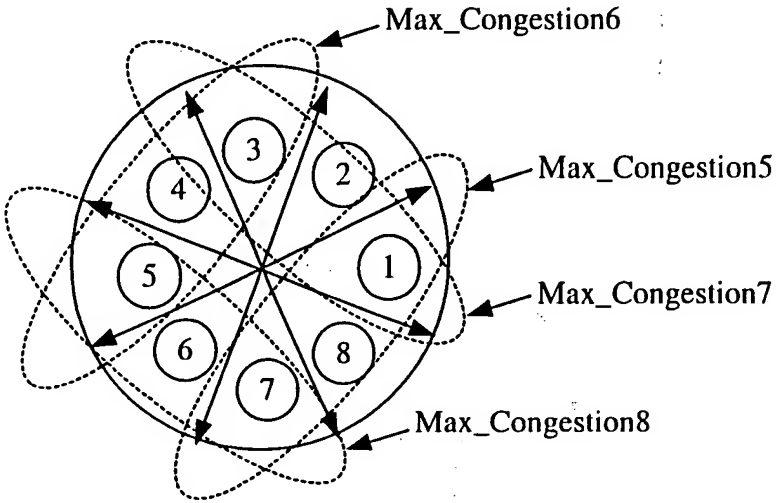


Figure 45

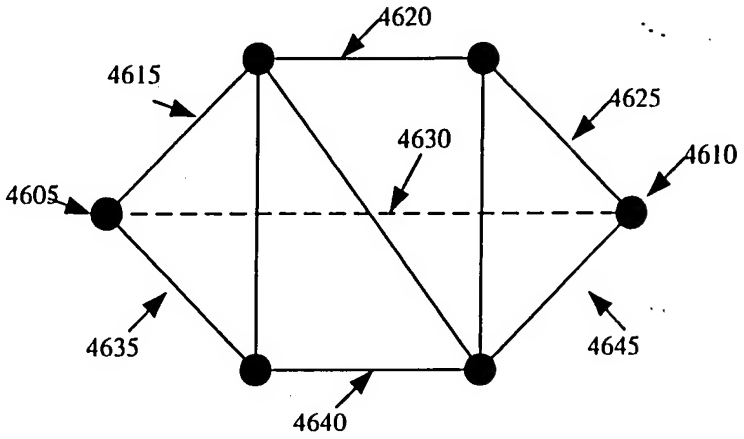


Figure 46

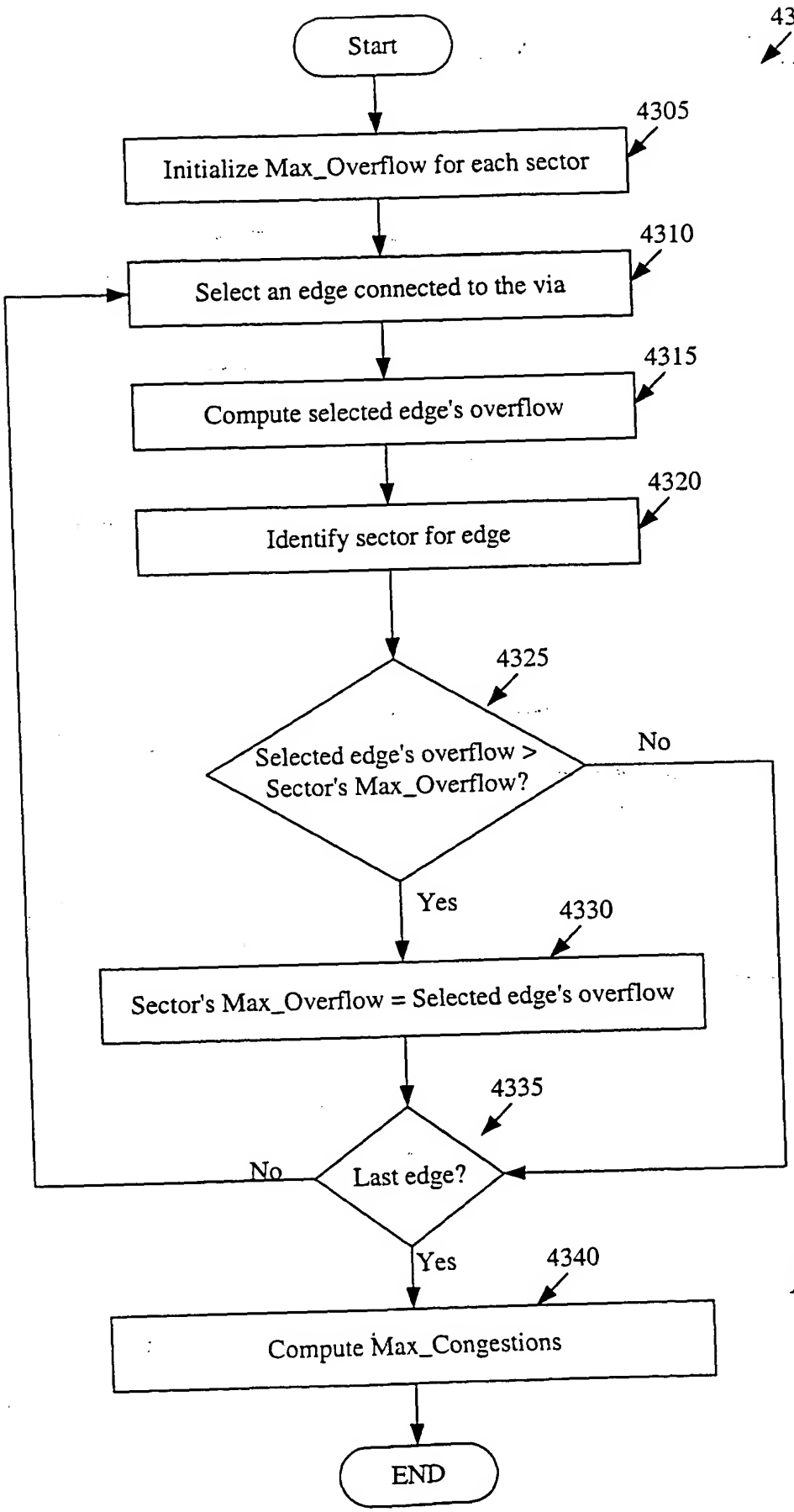

4300
Start
4305
Initialize Max_Overflow for each sector
4310
Select an edge connected to the via
4315
Compute selected edge's overflow
4320
Identify sector for edge
4325
Selected edge's overflow > Sector's Max_Overflow?
No
Yes
4330
Sector's Max_Overflow = Selected edge's overflow
4335
Last edge?
No
Yes
4340
Compute Max_Congestions
END

Figure 43

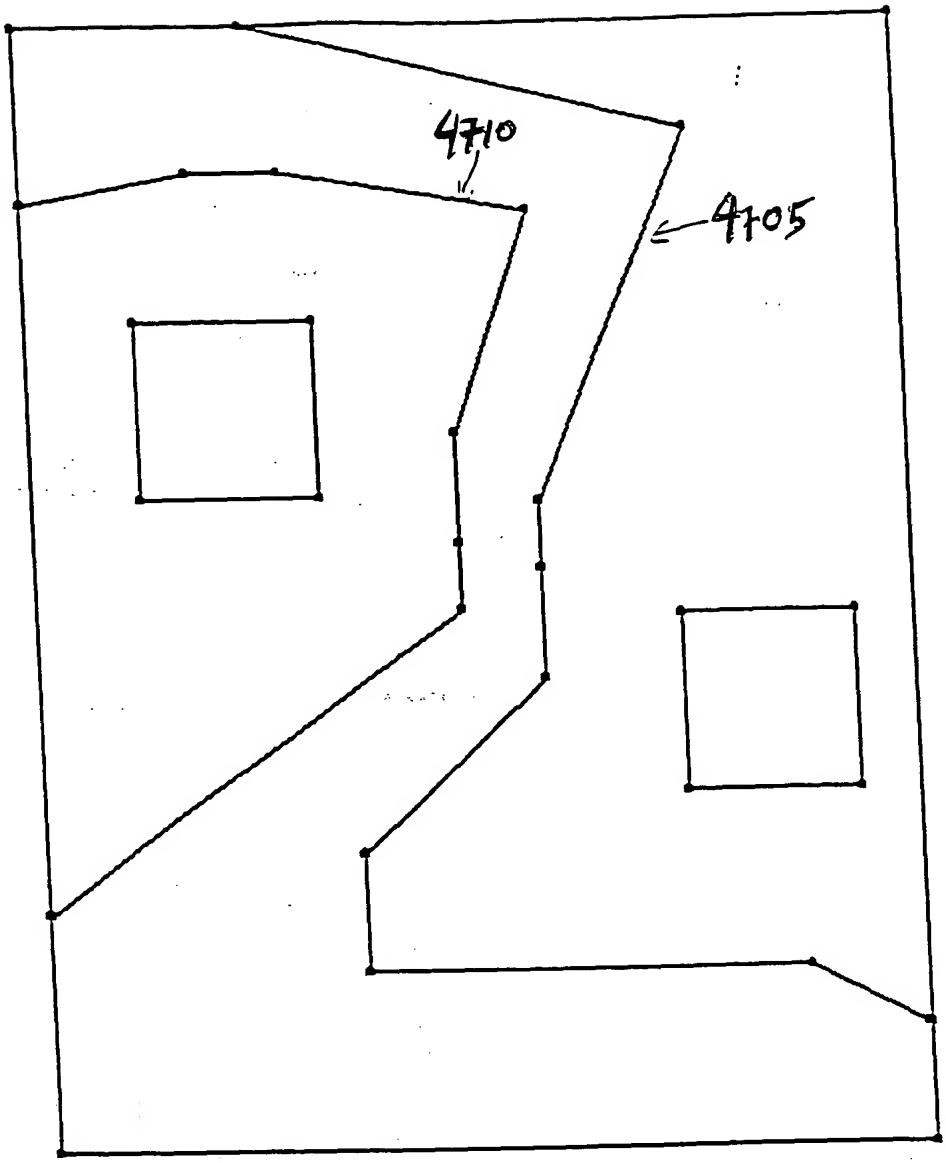


FIGURE 47

20120194502

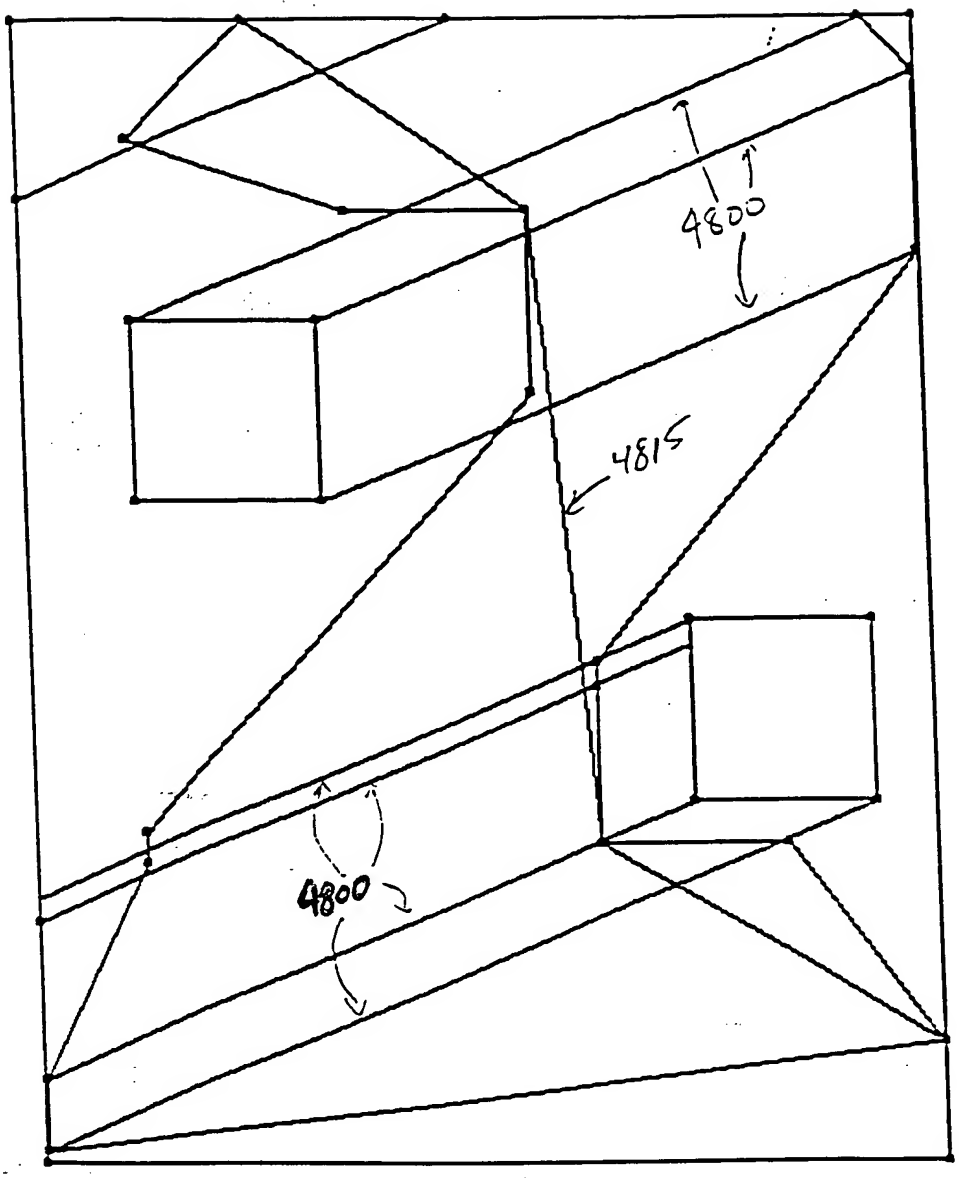


FIGURE 48A

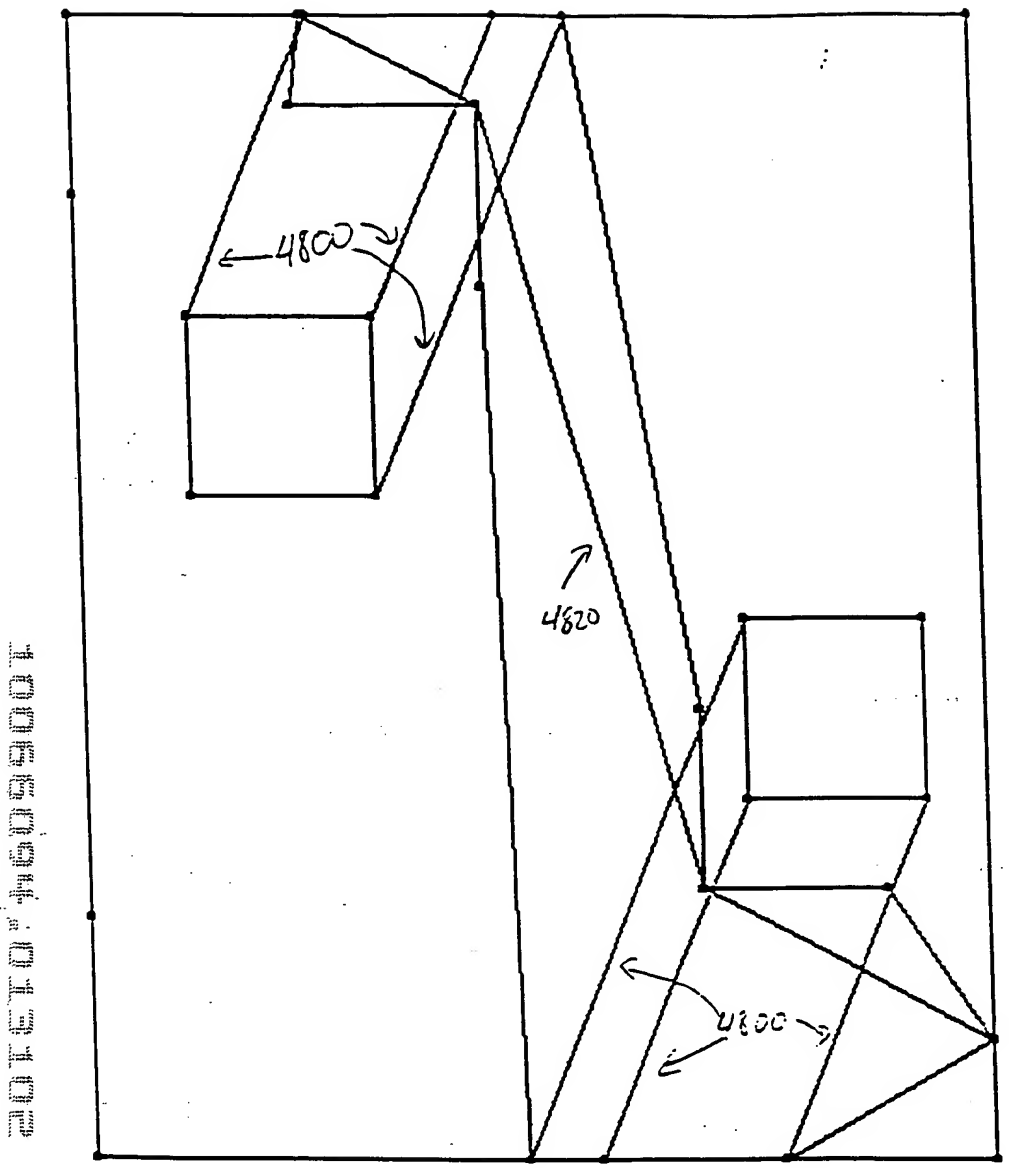


FIGURE 48B

20TC10-460990OT

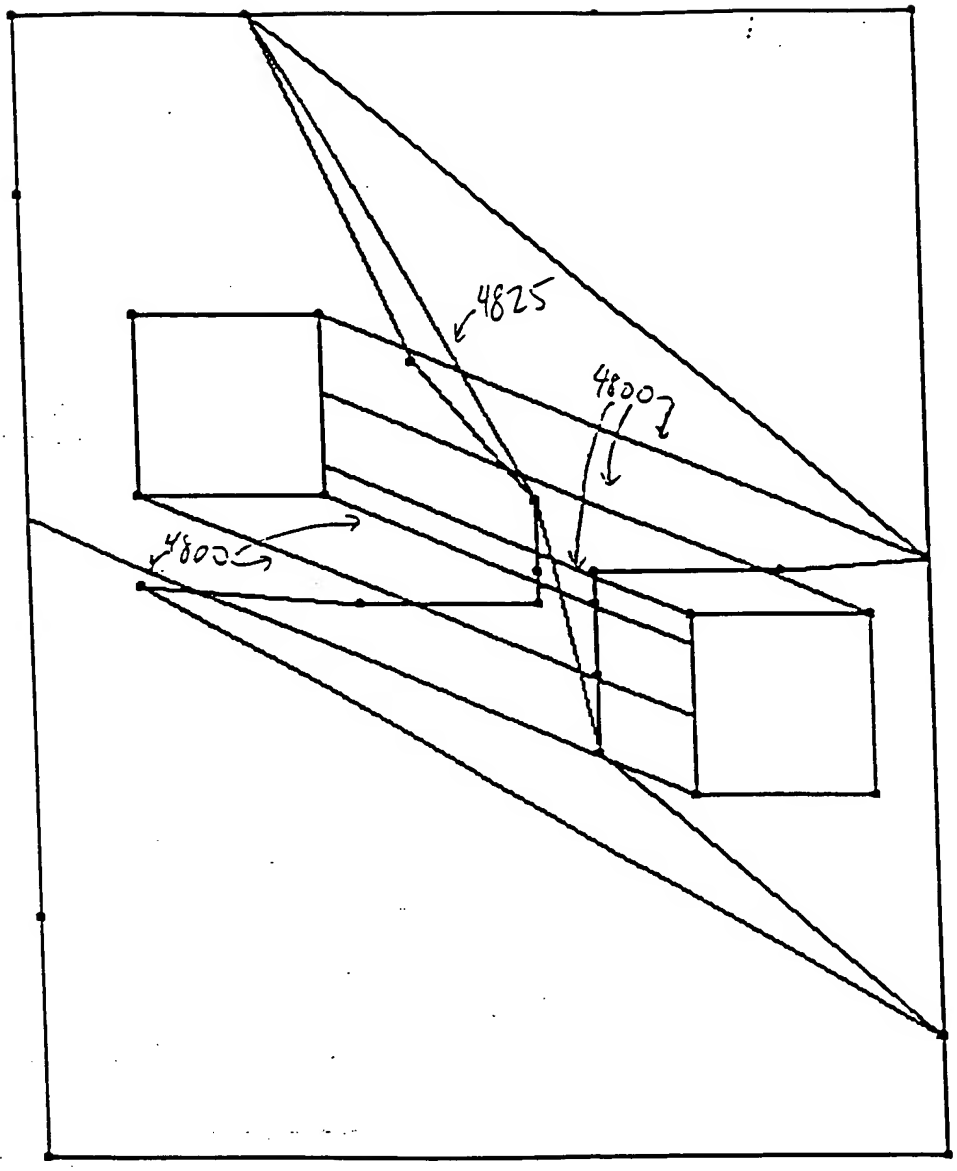


FIGURE 48C

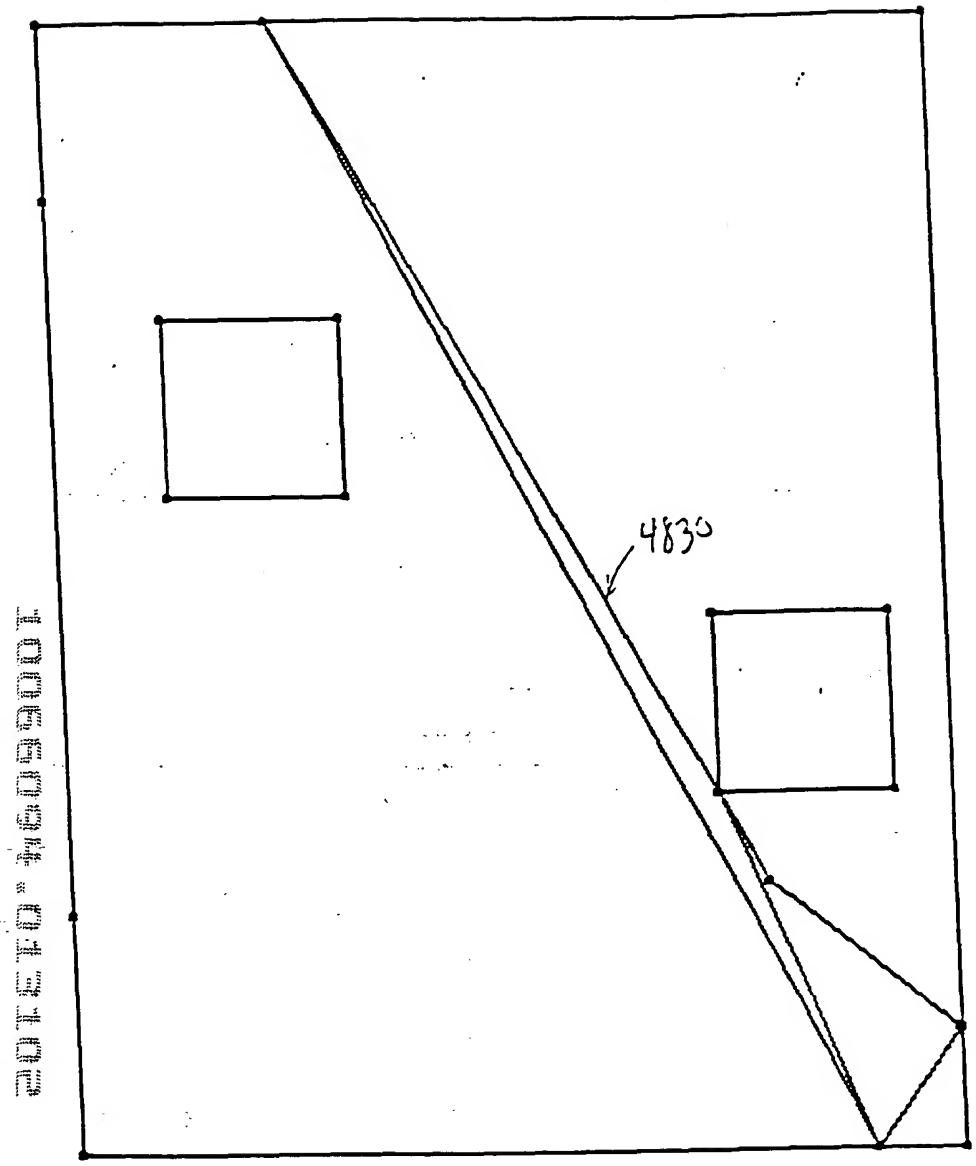


FIGURE 48D

20150"460990DT

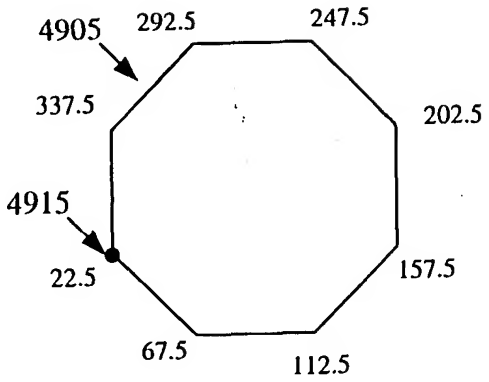


Figure 49A

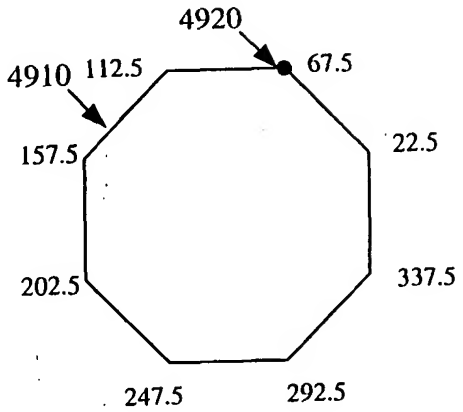


Figure 49B

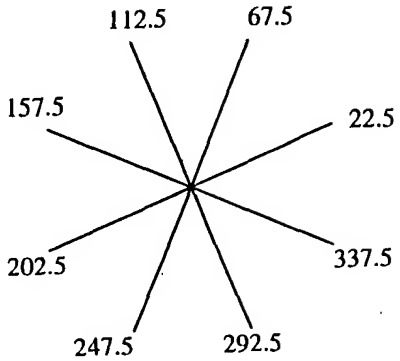


Figure 49C

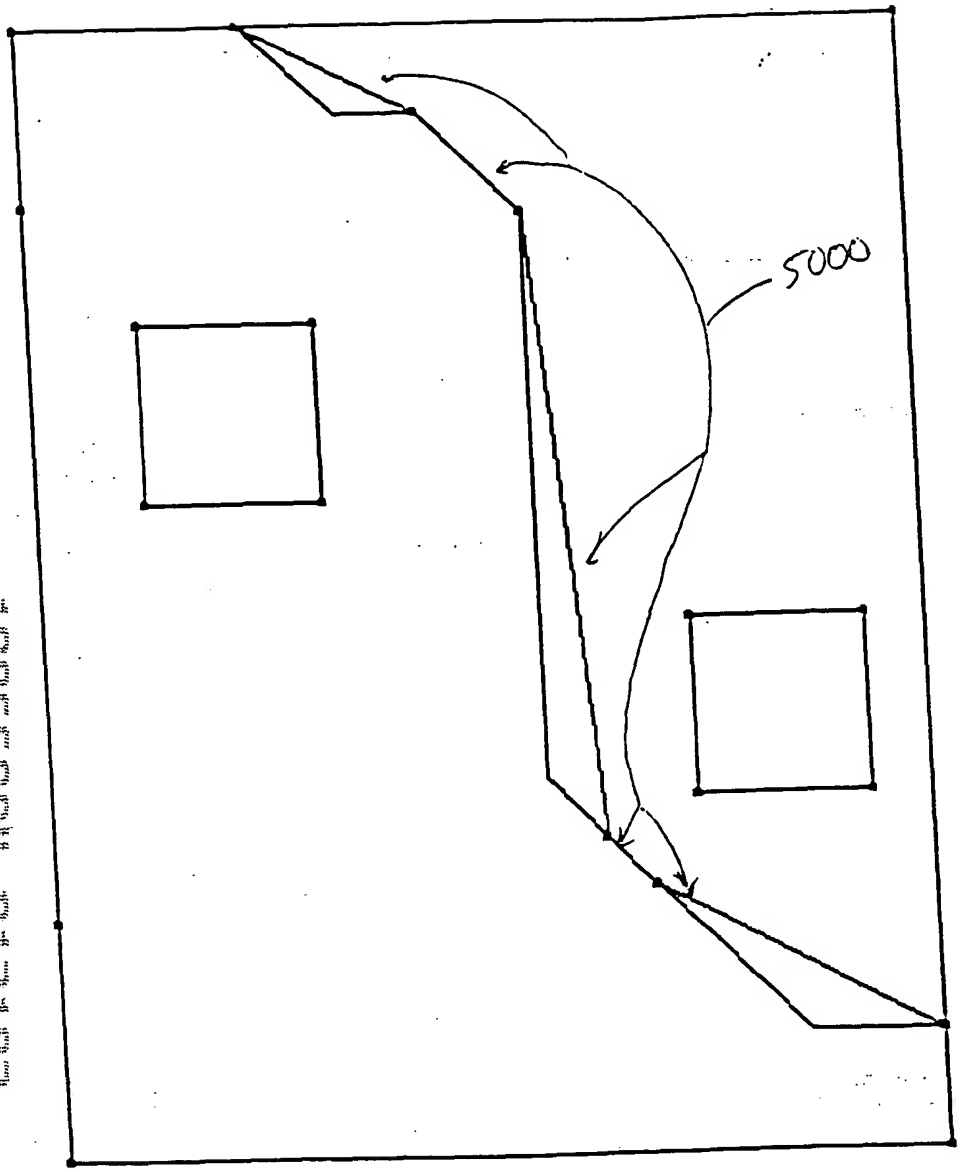


FIGURE 50

20020"A609900T

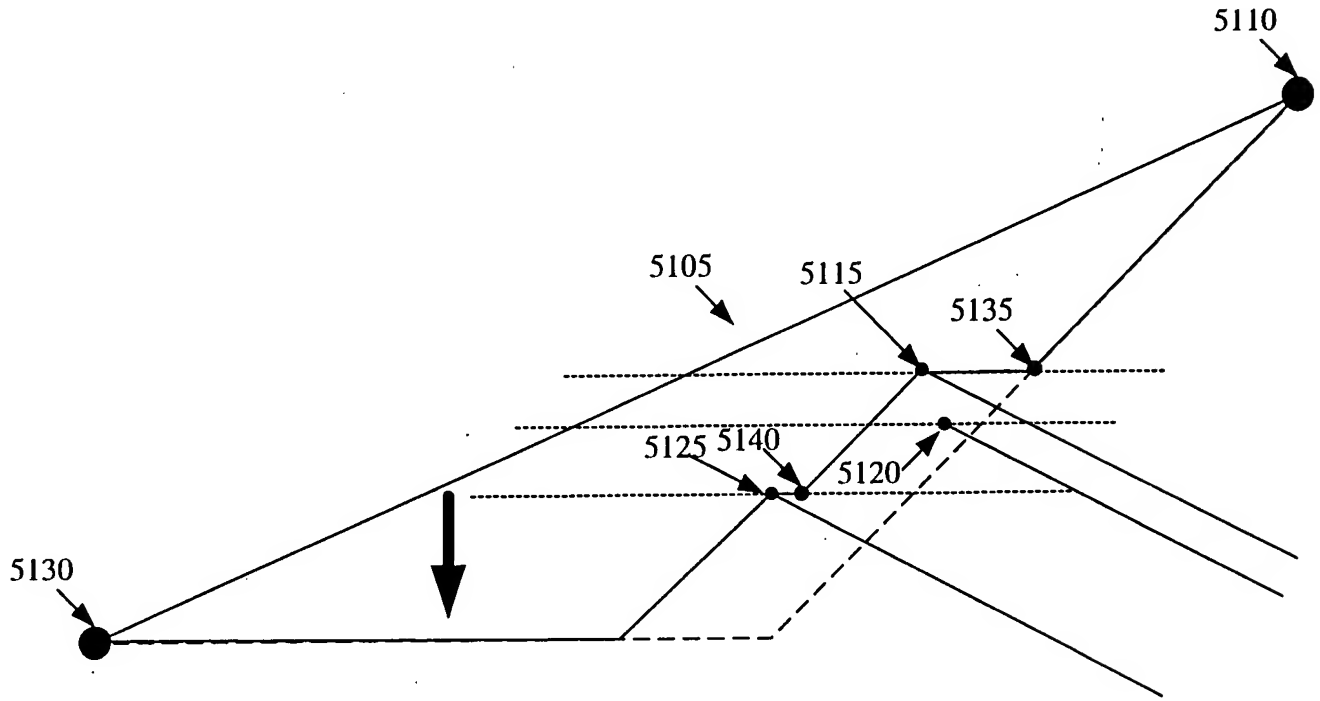


Figure 51

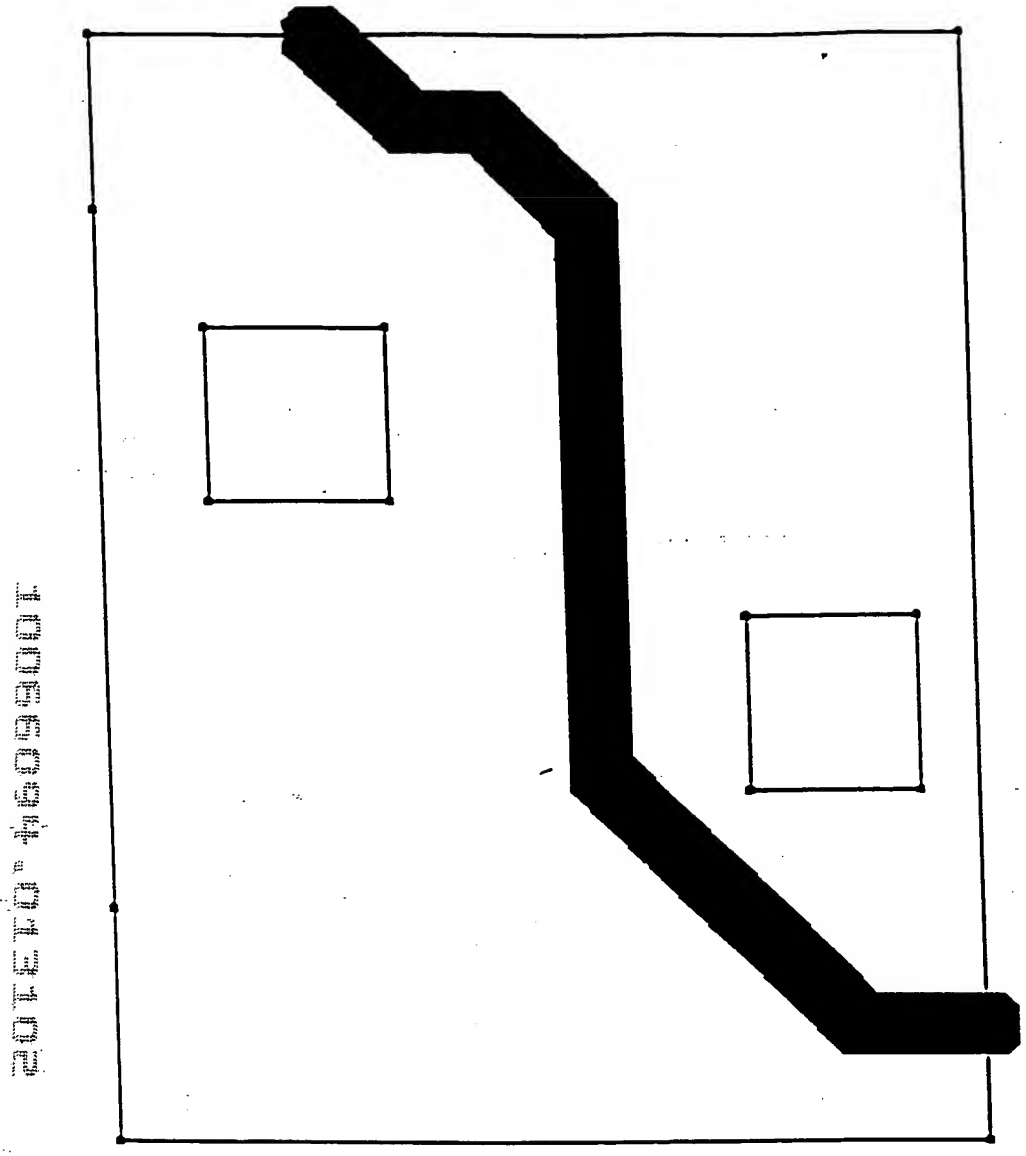


FIGURE 52

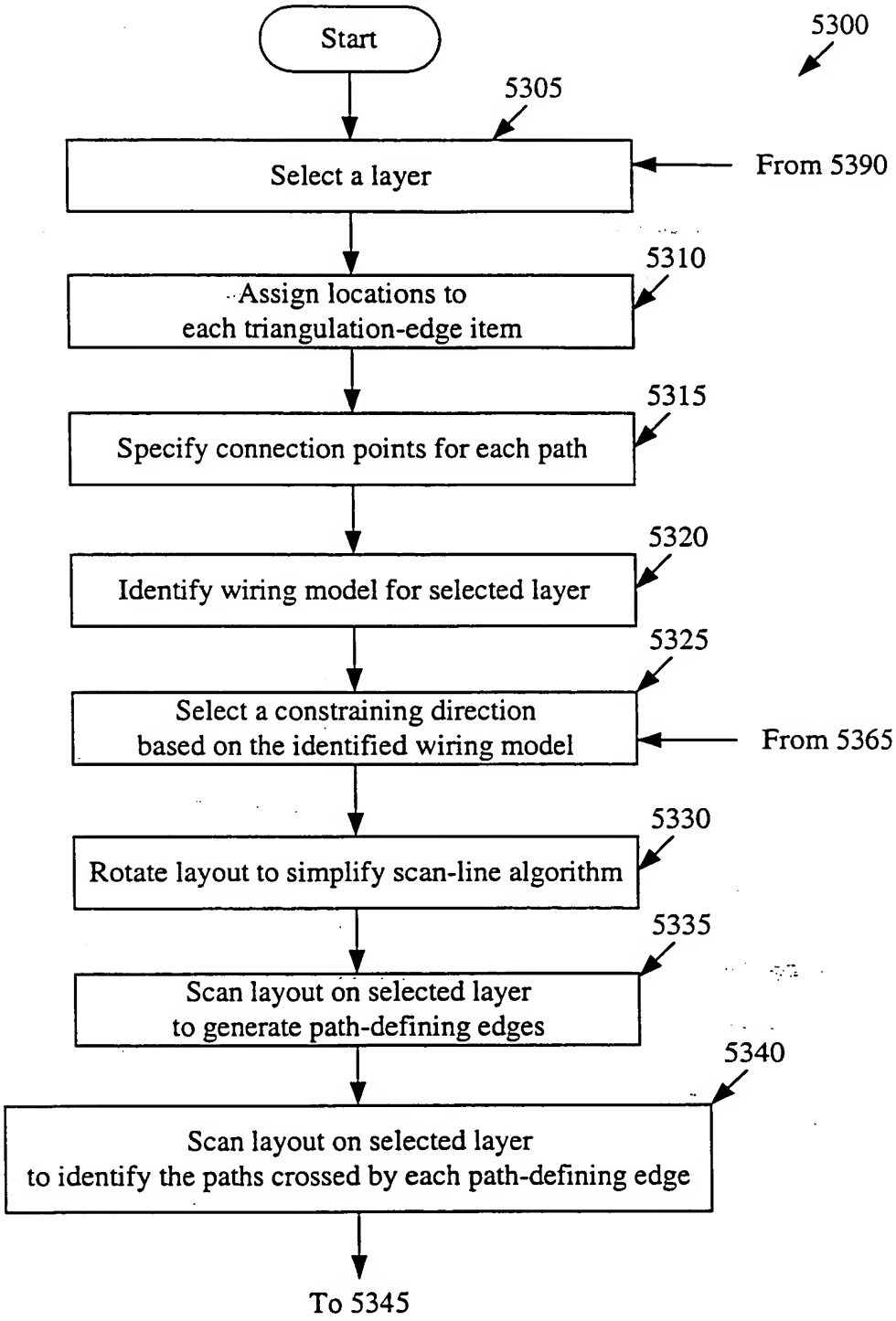


Figure 53

Figure 53: Figure 53A / Figure 53B

20140204500A1

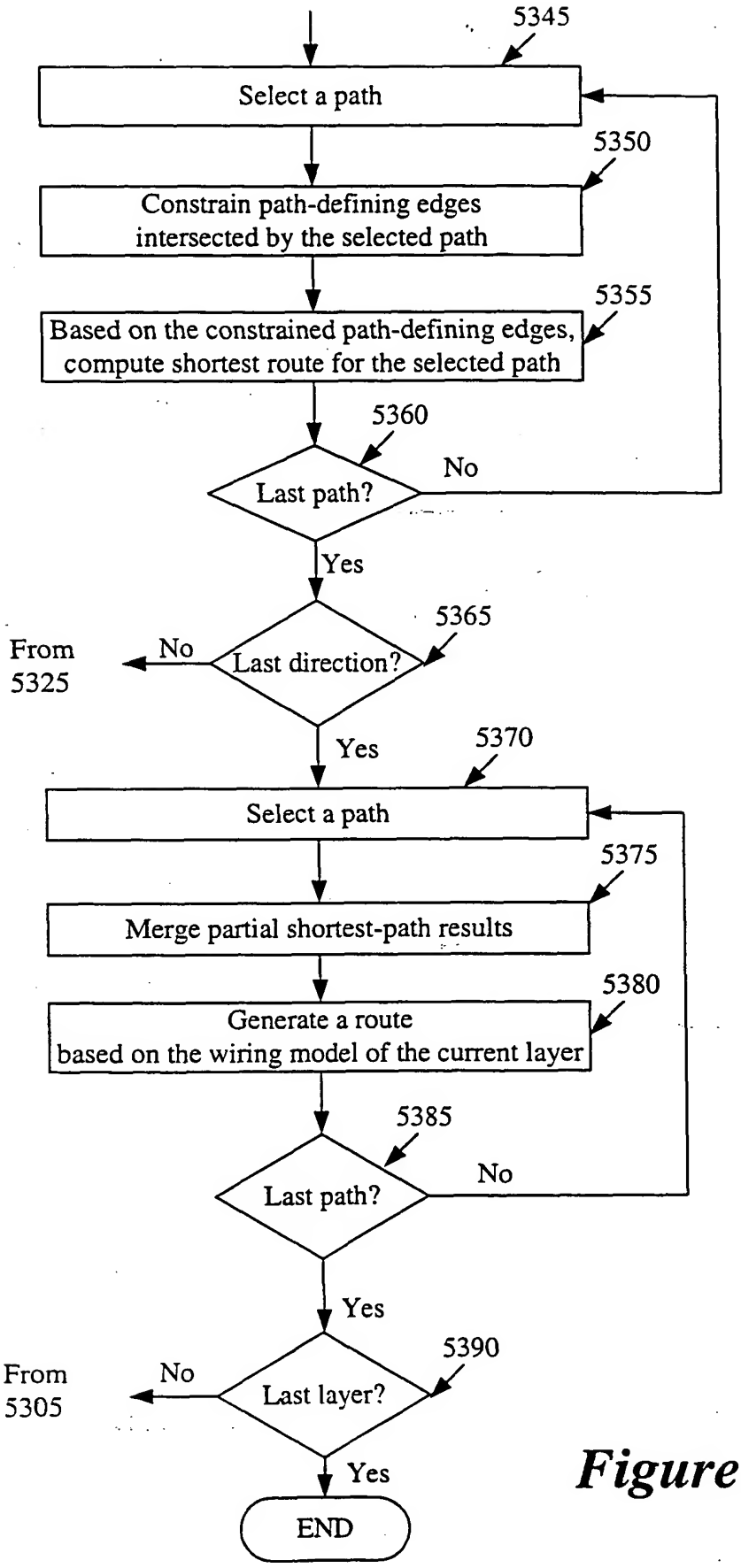


Figure 53B

20120"h6099001

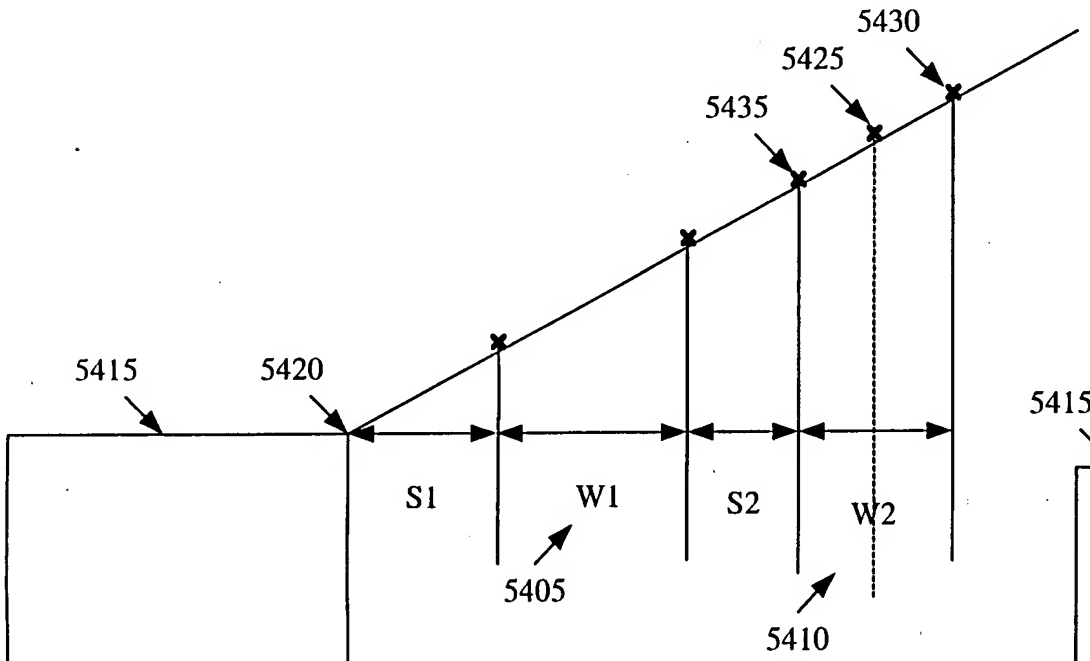


Figure 54

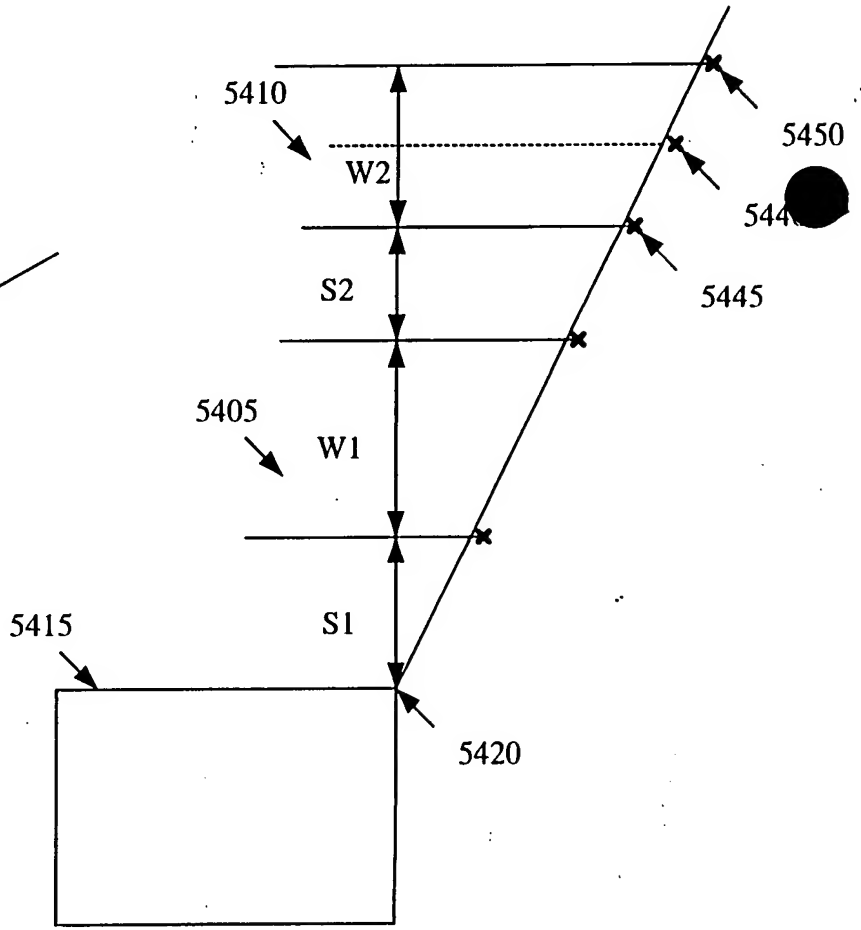


Figure 55

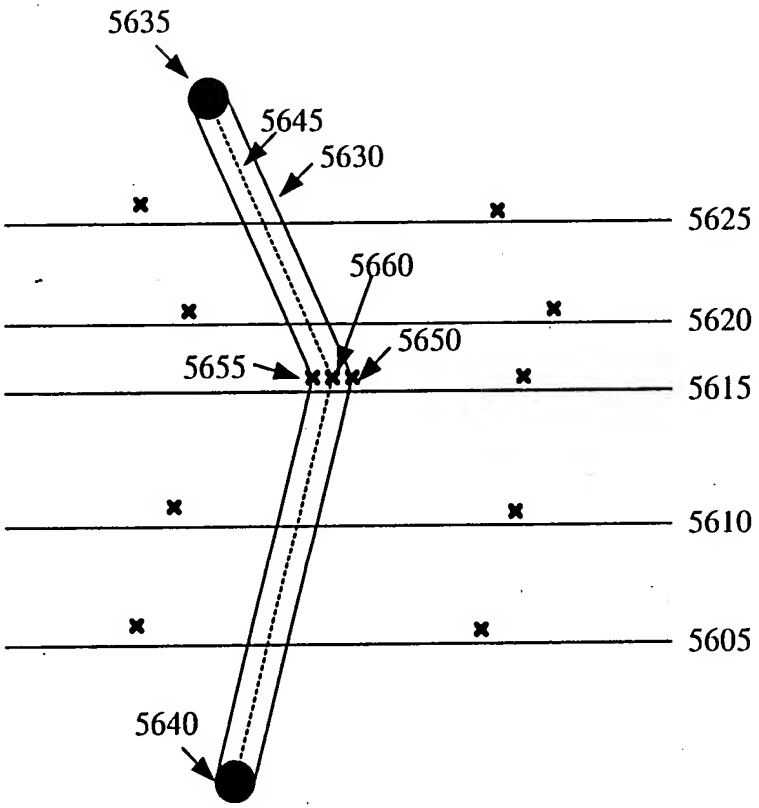


Figure 56

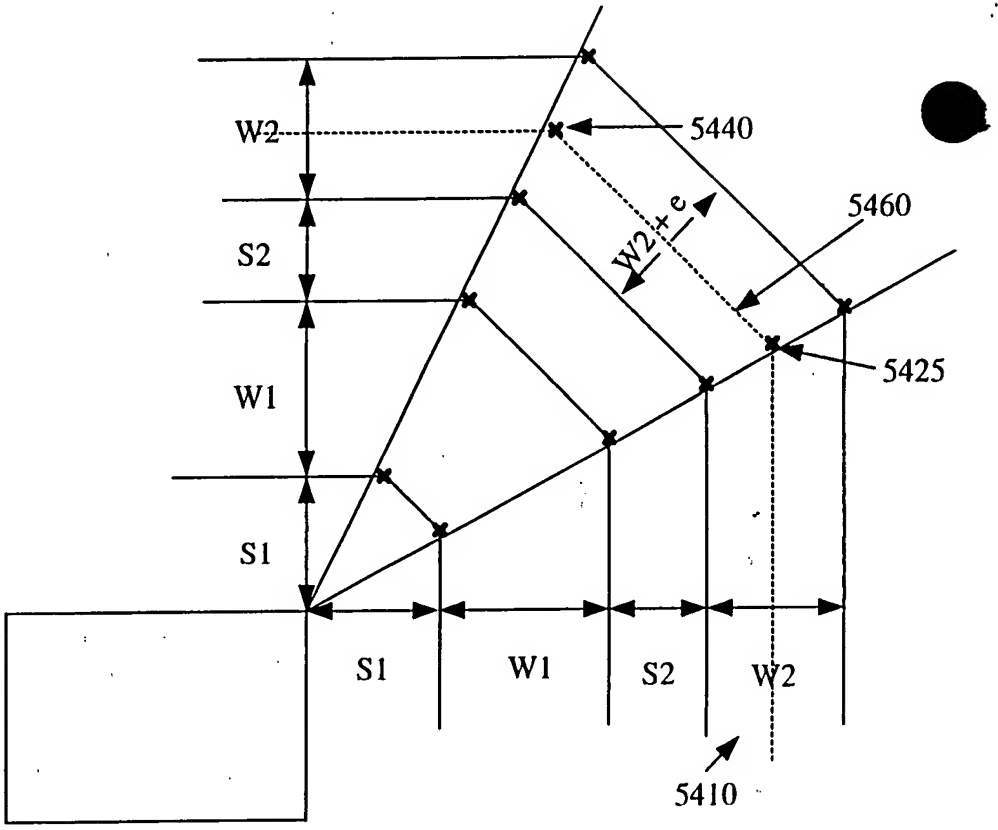


Figure 57

20TETO"W6099001

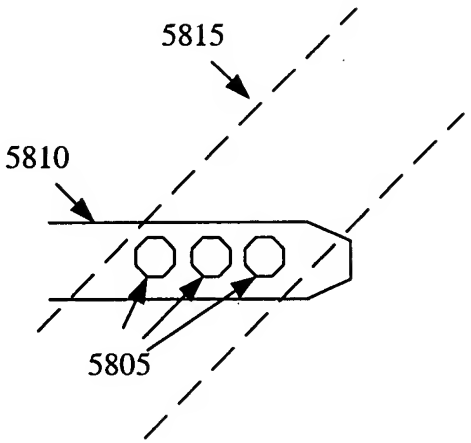


Figure 58

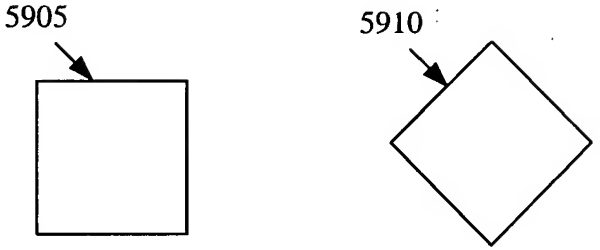


Figure 59

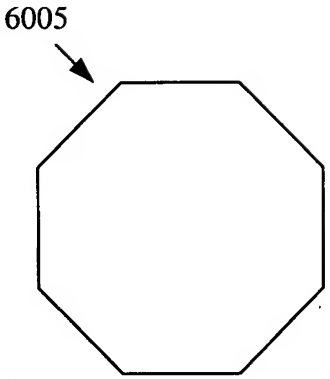


Figure 60

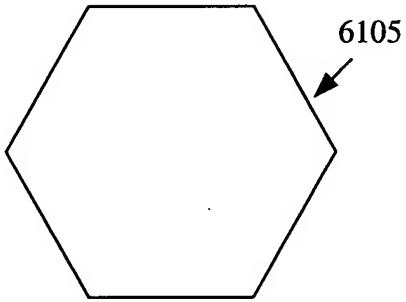


Figure 61

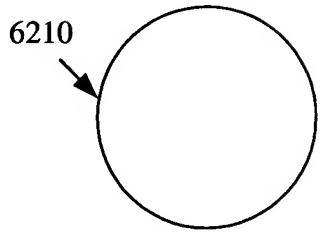


Figure 62

20EE0 "h609900I

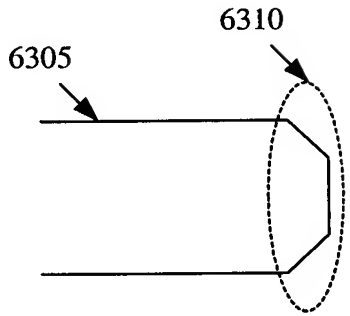


Figure 63

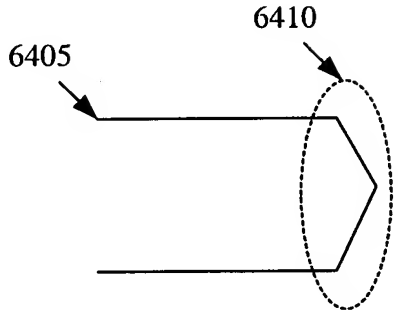


Figure 64

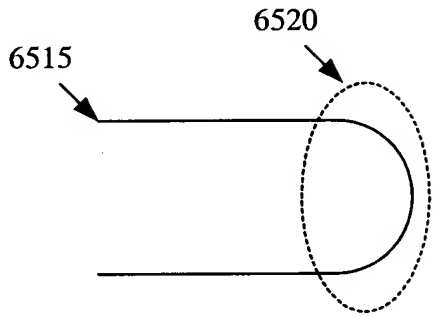


Figure 65

(1) *(2)* *(3)*

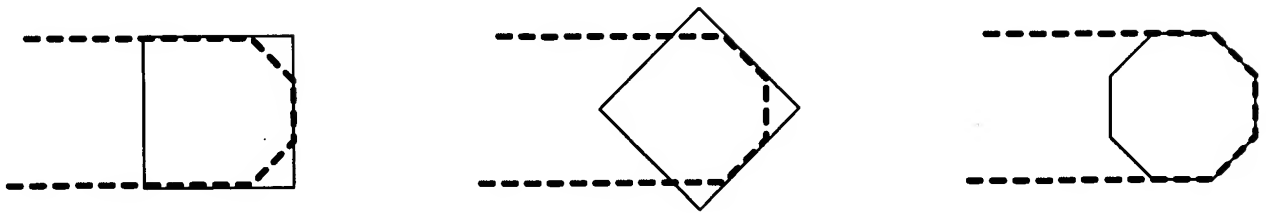


Figure 66

20TETO"h6099001

Figure 67

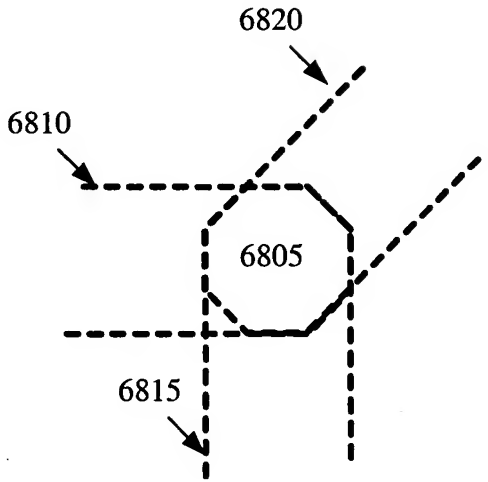


Figure 68

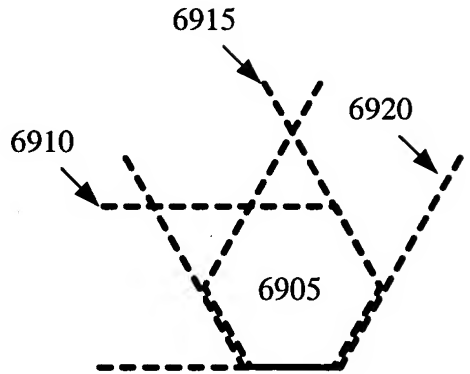


Figure 69

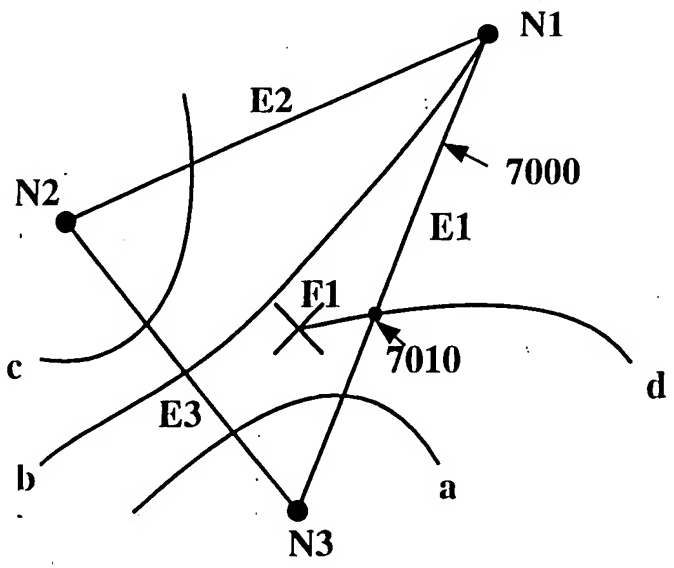


Figure 70

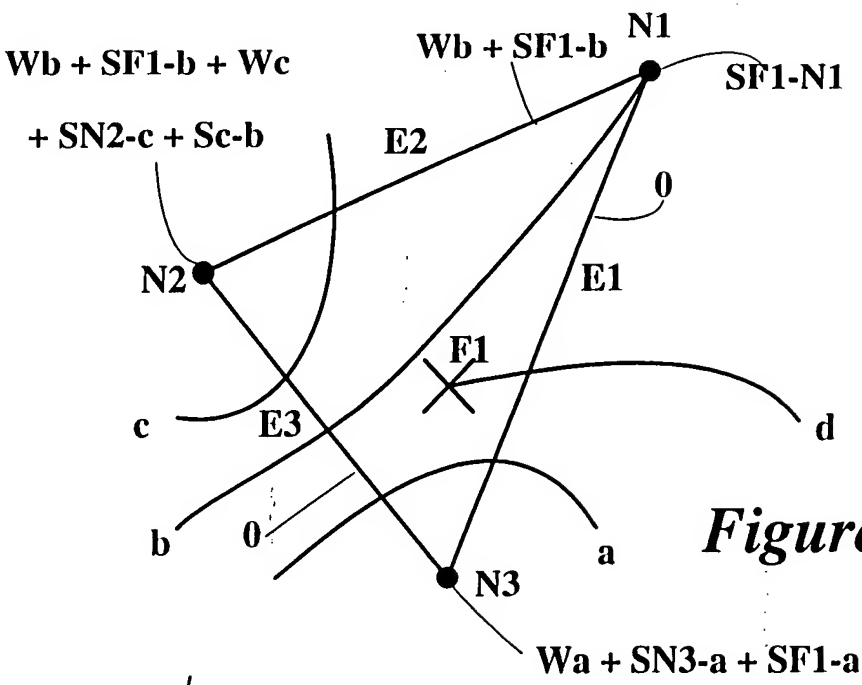


Figure 71

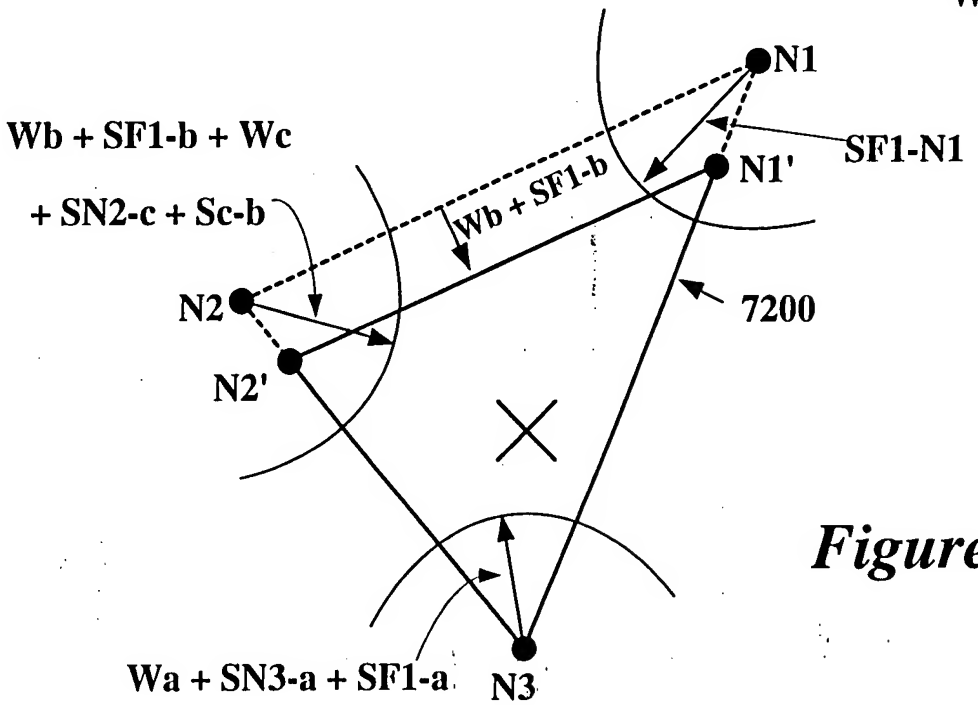


Figure 72

20TETO"h6099001

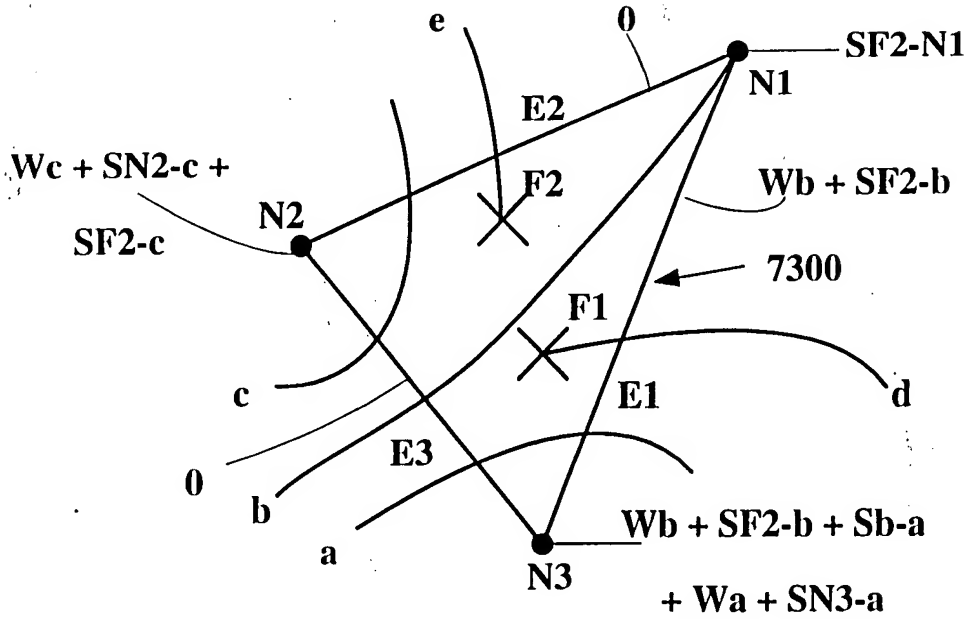


Figure 73

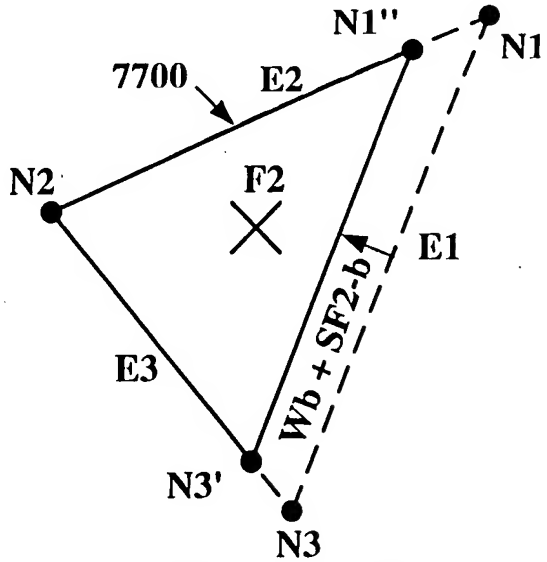


Figure 77

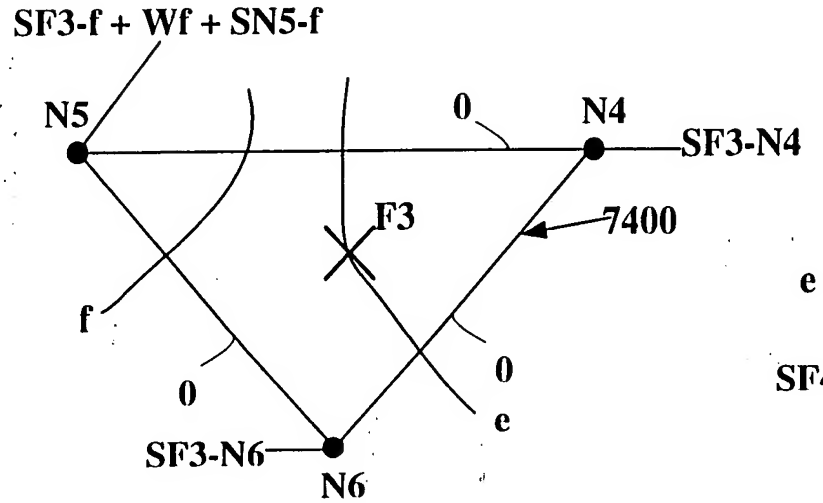


Figure 74

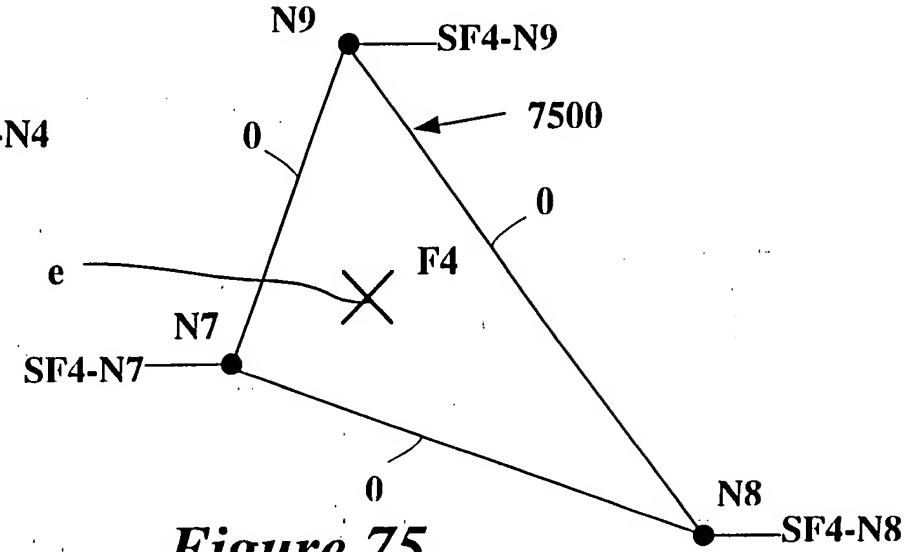


Figure 75

20TETO"h6099DOT

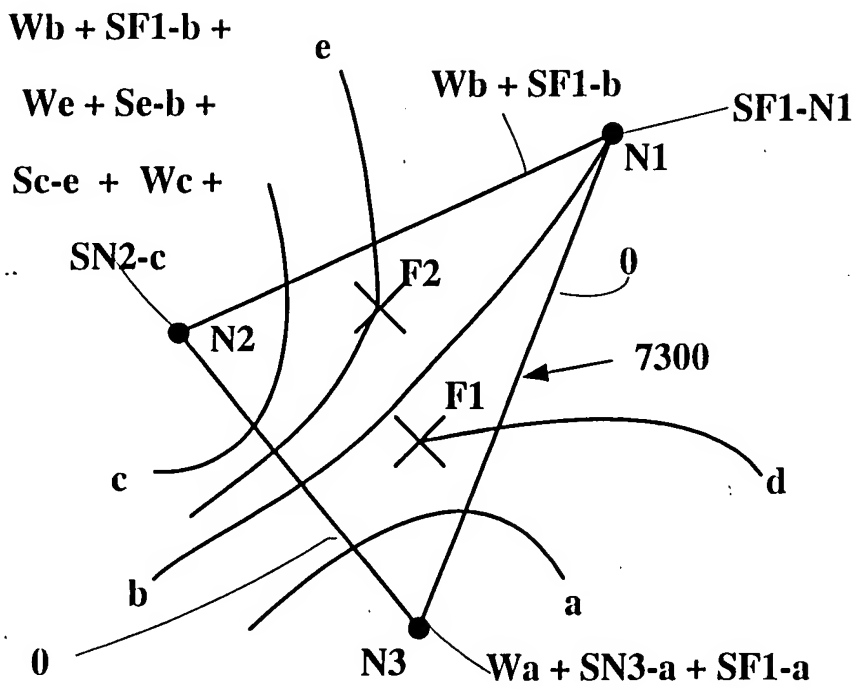


Figure 76

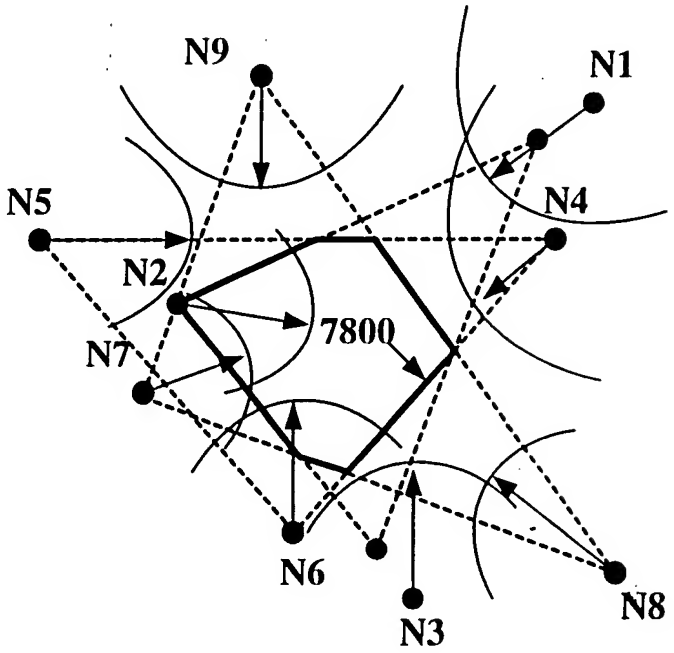


Figure 78

20TETO"4609900T

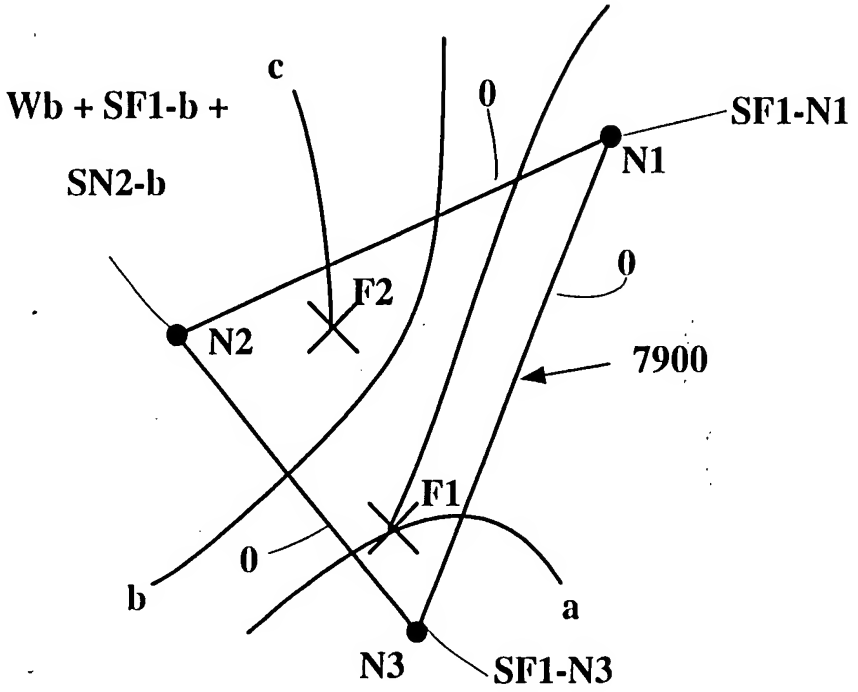


Figure 79

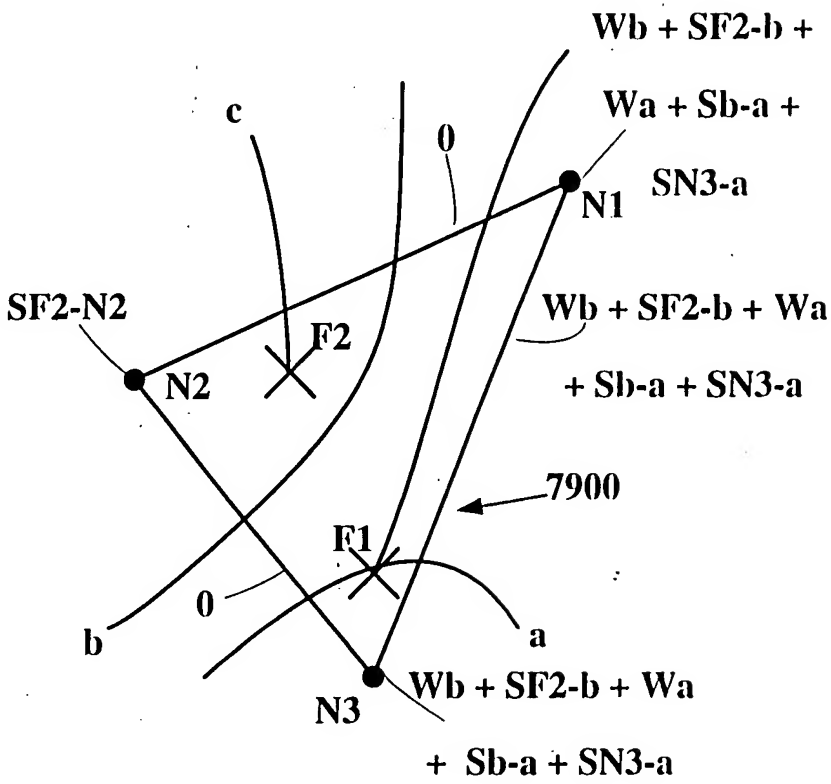


Figure 80

20TST0 "tB099DT

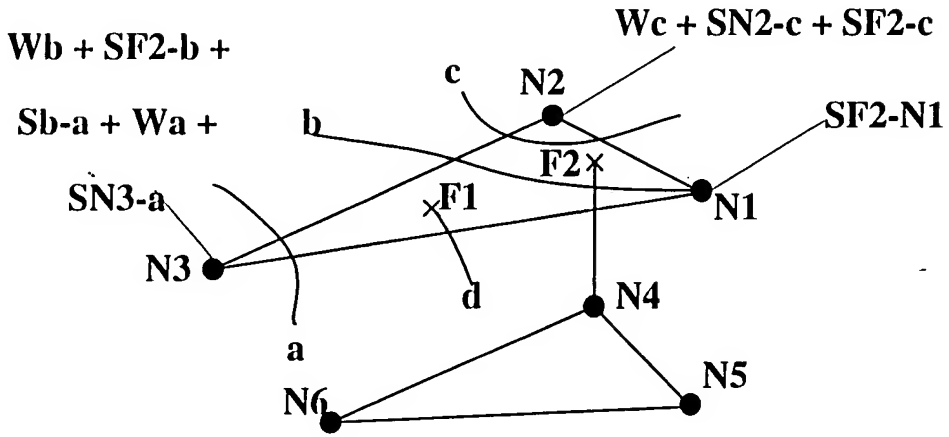


Figure 81

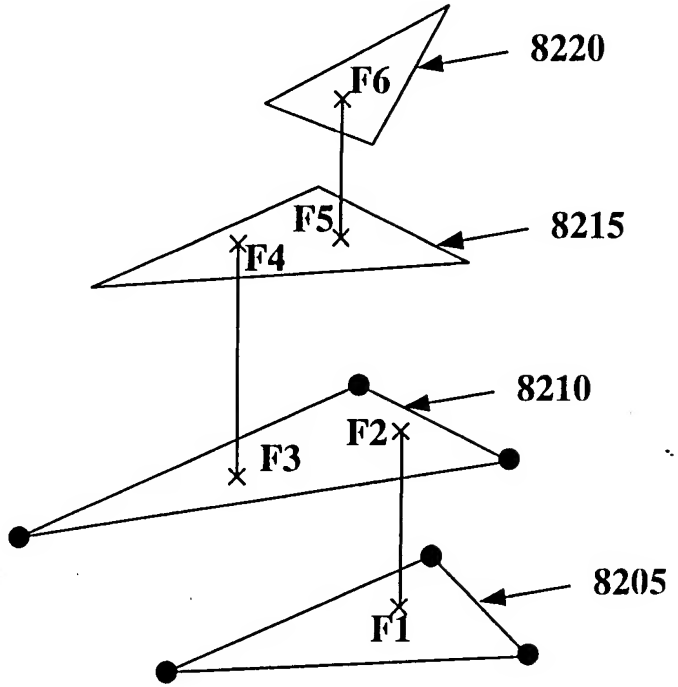


Figure 82

20TETO"4609900T

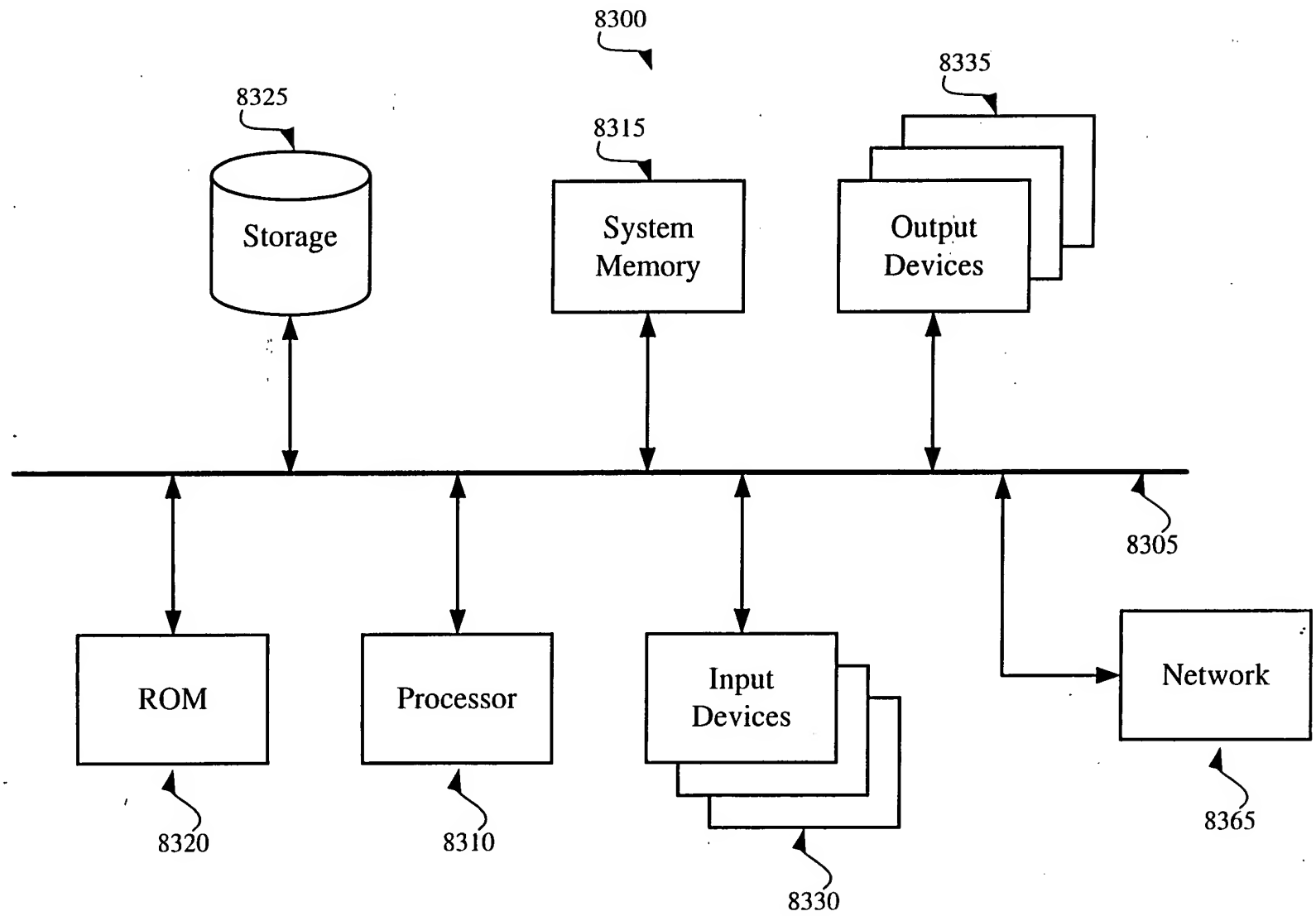


Figure 83